Heriot-Watt University

PhD Thesis

# Multimodal Headpose Estimation and Applications


*Author:*
Sankha Subhra Mukherjee

*Supervisor:*
Prof. Neil Robertson


*Submitted for PhD (Electrical)*

*in*


Institute of Sensors, Signals and Systems
School of Engineering and Physical Sciences


April 2017

# Declaration of Authorship

I, Sankha Subhra Mukherjee, declare that this this work submitted for assessment is my own and expressed in my own words. Any uses made within it of works of other authors in any form (e.g., ideas, figures, text, tables) are properly acknowledged at their point of use. A list of the references employed is included.

Signed:

Date: April 2017

# *Abstract*

This thesis presents new research into human headpose estimation and its applications in multi-modal data. We develop new methods for head pose estimation spanning RGB-D Human Computer Interaction (HCI) to far away "in the wild" surveillance quality data. We present the state-of-the-art solution in both head detection and head pose estimation through a new end-to-end Convolutional Neural Network architecture that reuses all of the computation for detection and pose estimation. In contrast to prior work, our method successfully spans close up HCI to low-resolution surveillance data and is cross modality: operating on both RGB and RGB-D data. We further address the problem of limited amount of standard data, and different quality of annotations by semi supervised learning and novel data augmentation. (This latter contribution also finds application in the domain of life sciences.)

We report the highest accuracy by a large margin: 60% improvement; and demonstrate leading performance on multiple standardized datasets. In HCI we reduce the angular error by 40% relative to the previous reported literature. Furthermore, by defining a probabilistic spatial gaze model from the head pose we show application in human-human, human-scene interaction understanding. We present the state-of-the art results on the standard interaction datasets. A new metric to model "social mimicry" through the temporal correlation of the headpose signal is contributed and shown to be valid qualitatively and intuitively. As an application in surveillance, it is shown that with the robust headpose signal as a prior, state-of-the-art results in tracking under occlusion using a Kalman filter can be achieved. This model is named the Intentional Tracker and it improves visual tracking metrics by up to 15%.

We also apply the ALICE loss that was developed for the end-to-end detection and classification, to dense classification of underwater coral reefs imagery. The objective of this work is to solve the challenging task of recognizing and segmenting underwater coral imagery in the wild with sparse point-based ground truth labelling. To achieve this, we propose an integrated Fully Convolutional Neural Network (FCNN) and Fully-Connected Conditional Random Field (CRF) based

classification and segmentation algorithm. Our major contributions lie in four major areas. First, we show that multi-scale crop based training is useful in learning of the initial weights in the canonical one class classification problem. Second, we propose a modified ALICE loss for training the FCNN on sparse labels with class imbalance and establish its significance empirically. Third we show that by artificially enhancing the point labels to small regions based on class distance transform, we can improve the classification accuracy further. Fourth, we improve the segmentation results using fully connected CRFs by using a bilateral message passing prior. We improve upon state-of-the-art results on all publicly available datasets by a significant margin.

# Contents

# List of Figures

# Chapter 1

# Introduction

Our society is currently faced with multiple paradigm shifting ideas. On one hand, with ever increasing population in ever growing urban spaces, security and maintaining law and order has become of paramount importance. The number of surveillance cameras, both in public and private spaces have exploded to unprecedented numbers in the last decade. Along with advancements in storage, affordable sensors and other internet services like Youtube$^{TM}$, this has lead to the proliferation of High Definition and Ultra High Definition videos in all aspects of life capturing the human experience from innumerable different perspectives. This presents new opportunities as well as new challenges. The rich information source present in the data calls for more advanced analytic methods. One of the prime component of this is understanding and predicting human behaviour. Not only who they are but who/what they interact with and what they are about to do. All of this information, along with long term tracking, allows us to build very strong models for people and their behaviours that might find applications ranging from security to targeted advertisement.

On the other hand, new specialised fields like advanced robotics and self driving cars present another unique set of problems, that are ripe for innovative solutions. Often these problem domains involve multiple modalities like depth, audio etc. alongside videos. Understanding human behaviour in a scene is very important for self driving cars faced with the decision of whether a human that came in to view suddenly saw it approaching or not. Is she going to cross the road? Or is she just standing on the pavement waving to a friend on the other side? Similarly for factories with heavy industrial robots and humans working together, how does

one ensure health and safety? Clearly for future autonomous systems to coexist with humans and share the same space, these challenges need to be adequately addressed and solved. To this end we believe, understanding human behaviour is going to be key.

To understand human behaviour, it will be important to understand subtleties of human gesticulation. In his pioneering work on psycholinguistics, Kendon has identified the salient features of human gesticulation [6]. While the content of a gesture varies from from person to person and even culture to culture, there are things that are invariant. One such invariant is that gestures are always communicative. That means, gestures always have a recipient, and people tend to look towards their recipient. Another such invariant is social mimicry, which means people in a group tend to behave similarly.

It is apparent from psycholinguistic works [6] that friends tend to look at each other when experiencing both duress and fun among a group of strangers, a very good cue predicting association. Groups of people in an exhibition tend to follow similar gazing patterns. Moreover it is intuitive that people tend to look where they are going before changing course.All this means that one of the corner stones to understanding the visual aspects of human behaviour is human head pose.

## 1.1 Motivation

Modelling human head pose is a challenging problem in computer vision and signal processing. It is desirable because this headpose signal gives us important meta-information about communicative gestures [6], salient regions in a scene based on focus of attention [7], group detection, crowd behavioural dynamics and tracking [8], and anomaly detection. The grand aim of our work is to exploit the advanced signal acquired from head pose to achieve, what is called, "Social Signal Processing". In domains where close level iris/eye tracking is not possible, human head pose is the most important feature in estimating human focus-of-attention. Head pose estimation has been studied in two separate and distinct domains, visual surveillance [9–12] and Human Computer Interactions (HCI) [13–15] with different methodologies required due to the difference in the quality of the input. In this work we develop a new technique which unifies these research areas and exploits the multiple modalities of range images and colour images when it is beneficial so

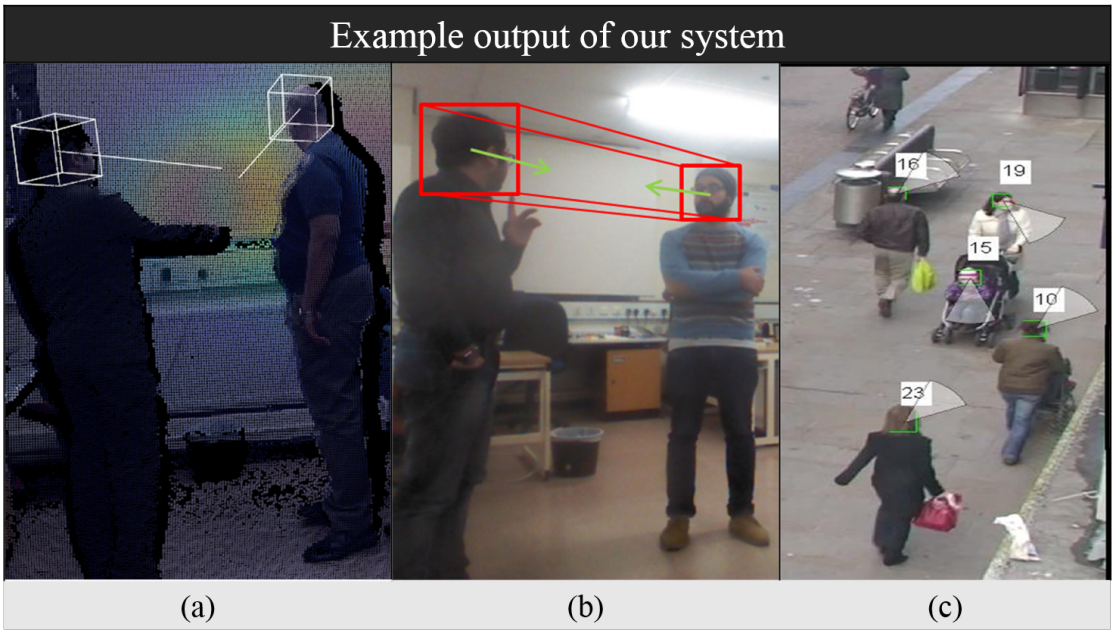



FIGURE 1.1: Illustrative outputs of our system showing gazing direction estimation across a range of imaging modalities, resolutions and applications: (a) visual attention modelling in the 3-D environment; (b) human-human and human-machine interaction recognition; (c) gazing direction in low-resolution surveillance video, which may ultimately be used for tracking and anomaly detection. sensor.

to do. The method is also highly robust and fast to compute as we demonstrate on data from both domains.

In the surveillance domain the nature of the problem requires the exploitation of priors such as walking direction [11] to augment the low resolution visual features. In the close range (i.e. higher resolution) domain of HCI, facial landmark detection approaches are employed for better accuracy [1]. However in HCI, the problem has been formulated with natural user interaction in mind, i.e., the user is always facing (near frontal) the sensor and is fairly close by (not more than 1-2 meters). Facial landmark based techniques typical of HCI cannot perform unconstrained head pose estimation at a distance. Furthermore, in most indoor interaction scenarios, the subjects are static and can be frequently occluded.

Hence the priors such as motion direction, body direction that are easily exploitable in a surveillance scenario may not be useful. Our focus in this thesis is to introduce a system that addresses these issues by estimating the unconstrained head-pose by using a unified approach. Figure 1.1 shows some illustrative outputs of our method.

Any image of the head can be used to estimate parameters in this manifold. Occlusions such as hair, accessories like glasses, and/or low resolution make accurate estimation difficult. However since pose has a very high variance in the feature space, and thus a large eigenvalue principal component, this should allow us to recover the head pose in a holistic manner that spans the range of HCI to surveillance. Furthermore, in the surveillance domain the techniques rely on motion priors like walking direction to smooth the head pose estimation [10, 11]. This is valid only because most people *tend to look where they are going*. The drawback of such smoothing is that the information of the head pose signal itself is attenuated. As shown by Baxter et. al in [8] the cases of actual interest are when people deviate from this behaviour (i.e. look somewhere else). This information could be useful for anomaly detection or improving tracking and should not be smoothed out by a prior simply in order to achieve a more accurate result because the datasets are biased with people looking the direction of walking most of the time. Similarly, in the HCI domain, most techniques rely on the detection of facial landmarks. This is a valid assumption given the use-case scenarios. However this leaves a large gap in the applicability of such methods when it comes to achieving a reliable head-pose estimation in close range for non-frontal head pose. In summary, we define our objectives as follows:

1. Establish and review current methods;
2. Bridge the gap in data requirements if any;
3. Exploit high-resolution to low-resolution imageries and exploits multiple imaging modalities, i.e. RGB and depth where possible;
4. Is independent of explicit facial landmark detection;
5. Do not require motion priors ("instantaneous", i.e. only requiring single frames);
6. Create unified end-to-end detection and headpose estimation framework;
7. Evaluate against public datasets;
8. Show applicability of robust headpose estimation to social signal processing;

We categorically don't use motion prioirs to decouple the headpose signal from the velocity signal. This preserves the two signals independently and does not attenuate the raw headpose signal with a prior.

Our thesis is as follows: *A big data driven machine learning approach can be adapted to solve the human head pose estimation problem. By using modern supervised and unsupervised machine learning techniques we can solve the problem of detecting heads and estimating head pose in an unified way that spans multiple modalities (RGB and Depth) while being applicable to high resolution Human Machine Interface to low resolution surveillance domain simultaneously. The robust headpose estimation can then be used a signal for Social Signal Processing. We evaluate the success of our approach by reporting higher accuracy compared to exsisting techniques on publicly available datasets.*

## 1.2 Thesis Roadmap

We have identified that there is a gap in the landmark free head-pose estimation research. We aim to contribute a solution when it comes to unconstrained head-pose estimation in all resolutions. Our approach to solving this problem is summarised as follows

In Chapter 2 we review the relevant research in head pose estimation. First we review the existing solutions and find out what limitations are there that we need to overcome. Then we discuss the current state-of-the-art methods in data driven machine learning approaches, mainly deep learning. We identify areas that in both detection and pose estimation that can be improved.

In Chapter 3 we identify the need for an unified dataset that is lacking in literature. We propose a large scale RGB-D dataset that has accurate headpose annotation. We also establish the theoretical limits of head pose estimation accuracy when eyes are not tracked. We then build upon current state-of-the-art methods and define and validate our own RGB-D feature called Histogram of Depth Surface Curvature (*HDSC*). We propose to use the novel Zonal Kernel for gaussian process regression for regression in a closed circular manifold for head yaw angles. We also propose to smooth the temporal estimations with particle filters. We compare our method to the existing methods on publicly available and our own datasets.

In Chapter 4 we embark upon the deep learning methods. First we propose a semi supervised model that can learn from data sources that are labelled differently. We propose a multi-loss fine tuning. We validate the approach on two publicly available surveillance RGB datasets.

In the next section we strive to model the human heads using Convolutional Neural Networks in both RGB and Depth. We introduce a regression loss that lets us pose the cyclic function (it is wrapped in a sphere) in the Euclidean space by using vector decomposition of the unit directional vector. We also model the regression confidence by a granular classification layer that only learns from the final layer feature used for the regression. We validate our approach on 6 publicly available datasets.

In the final section we propose an efficient fully convolutional RGB neural network that does detection and pose estimation, while sharing all of the computation among both tasks. We propose a novel ALICE loss function that jointly optimises both detection and pose estimation. We introduce a general region proposal network after the training to generate the head proposals after the whole computation through the network. We compare the detection and and pose estimation performance on two standard datasets.

We report significant improvements in accuracy all across the board with our methods. We prove that our methods are robust to resolution and modality while not being dependent on motion priors.

In Chapter 5 we apply the robust headpose estimation algorithm to other problems. First we show that we can improve tracking accuracy in presence of occlusion when using the head pose as an intentional prior. We then define three metrics (a) The Attention Metric (AM) that takes into account the direction and variance of the headpose estimation, (b) The Interaction Metric (IM) that predicts when people are looking at each other, and (c) Windowed Cross Correlation (WCC) between any pair of headpose signal that estimates how much two head pose signals behave similarly in time.
We validate the AM and IM against the other metrics proposed in literature and show state-of-the-art performance on publicly available datasets. Further more we qualitatively show three scenarios and discuss the effects of all the three metrics.

In Chapter 6 we show the general applicability of the ALICE loss proposed in Chapter 4 to an interesting problem in marine sciences. We apply deep learning methods to classify underwater coral images. We overcome the challenge of having very sparse labels and propose a technique to generate dense segmentation of

underwater coral imagery. We achieve the highest accuracy by a large margin, nearing expert annotation accuracy while reducing time required by a factor of 2000×.

## 1.3 Related Publications

Our work has been published in the following locations.

- Sankha S Mukherjee and Neil M Robertson. Deep head pose: Gaze-direction estimation in multimodal video. *IEEE Transactions on Multimedia*, 17(11): 2094–2107, 2015
- Sankha S. Mukherjee, Rolf H. Baxter, and Neil M. Robertson. Watch where you're going! - pedestrian tracking via head pose. In *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pages 573–579, 2016. ISBN 978-989-758-175-5. doi: 10.5220/0005786905730579
- Rolf H Baxter, Michael JV Leach, Sankha S Mukherjee, and Neil M Robertson. An adaptive motion model for person tracking with instantaneous head-pose features. *Signal Processing Letters, IEEE*, 22(5):578–582, 2015
- Sankha S Mukherjee, Rolf H Baxter, and Neil M Robertson. Instantaneous real-time head pose at a distance. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 3471–3475. IEEE, 2015
- Henry Lea-Ann, Sankha S Mukherjee, Neil M Robertson, Laurence De Clippele, and J. Murray Roberts. Deep corals, deep learning: Moving the deep net towards real-time image annotation. In *6th International Symposium on Deep-Sea Corals*. Harvard, 2016

# Chapter 2

# Related Work

In this chapter we introduce work by other authors that are most relevant to the field of study. We first discuss the prior work on human head pose estimation as has been reported in literature. Then review the contributions in deep neural network in recognition, detection, pose estimation, tracking and segmentation as these are the methods we have built upon in our work. We provide critique of the existing methods where necessary. We try to cover as much recent material as possible and avoid going into too much historical context to keep the review concise. We cite other works in context in the following chapters where needed.

## 2.1 Head pose estimation

The pioneering work on low resolution head pose estimation was proposed by Robertson and Reid [9] which used a detector based on template training to classify head poses in eight directional bins. This approach is heavily reliant on skin colour detection. Subsequently this template-based technique was extended to a colour invariant technique by Benfold et al. [10]. They proposed a randomized fern classifier for hair face segmentation for the template matching. This work was later improved upon by Siriteerakul et al. [20] using pair-wise local intensity and colour differences. However, in keeping with all template based techniques in head-pose estimation, these suffer from two major problems: first, it is non-trivial to localize the head in low resolution images; second, different poses of the same person may appear more similar compared to the same head-pose of different persons.

This led some researchers to propose representing head images in a different feature space that has more discriminatory property for head pose independent of persons. Non-linear regression approaches like Artificial Neural Networks [21, 22] and high-dimensional manifold based approaches [23, 24] try to estimate the head poses in a continuous range. Chen and Odobez [11] proposed the state-of-the-art method for unconstrained coupled head-pose and body-pose estimation in surveillance videos. They used multi-level Histogram of Oriented Gradients (HOG) [25] for the head and body pose features and extracted a feature vector for an adaptive classification using high dimensional kernel space methods. These techniques are quite general and do not depend on the heads being in near frontal poses unlike the HCI techniques. Nevertheless the high degree of error or uncertainties that arise from these methods, render them unsuitable for the tasks like fine grained human interaction or attention modelling.

On the other hand, on the HCI side of the problem the formulation is limited to 2 metre distance from the sensor along with near-frontal head-poses. An iterative closest point (ICP) based mesh fitting approach has been employed for head pose detection [14, 26]. In [27] the candidate head poses are rendered and matched to the input depth image and the 6 degree of freedom pose is solved by optimizing via particle swarm optimisation. Fanelli et al. [1] used a randomized patch based decision forest regression for head pose regression. Work on head pose regression for scene and human interaction understanding has been presented [28]. This work focuses on head-pose regression and interaction detection in 2D movie/ tv-series scenes. While it is quite robust, this approach is limited in that it only works with yaw angles of $\pm 90^{\circ}$. However it does not depend on motion priors or specific facial landmark detections. Recently, manifold based metric learning methods have been applied to head pose estimation [29]. In another approach to manifold learning the spherical nature of the view manifold of objects is used as a strong prior [30]. Another approach reported in [31] uses reflection symmetry information in covariant features extracted from Gabor features. Features derived from local directional quaternary patterns (LDQP) have been used in conjunction with linear SVM successfully in high resolution RGB data [32].

## 2.2 Deep learning

Recently, deep learning, especially CNNs have been shown to learn robust non-linear representations from input data and have been especially successful on images [33, 34] and audio [35]. This is in contrast to traditional computer vision pipelines where problem specific ad-hoc features like HOG [25] are extracted. These features would typically be used as input to machine learning framework such as support vector machines (SVM) to achieve classification or regression. The power of deep models lie in their ability to learn layers of non-linear transformations on the data [34]. The resurgence of these methods started with the successful introduction of a class of deep generative models called Deep Belief Networks (DBN) and their unsupervised training using Contrastive Divergence (CD) [36]. The power of a generative model, as shown by Tang et al. [37], lies in being able to reconstruct original images under noise or heavy occlusions [36]. CNNs [38] on the other hand are supervised, discriminative and have mostly surpassed the DBNs in terms of accuracy on large labelled datasets like the Imagenet [39].

CNNs have also been applied in the multimodal RGB-D domain. Lu [40] demonstrated early fusion of RGB-D channels and used transfer learning to initialise the weights of the green, blue and depth channels with filters learned from the depth channel. More recently it has been shown that this form of early fusion is not very helpful because the network can not propagate meaningful gradients across channels [41, 42]. Hence RGB-D networks are generally trained with late fusion where the modalities are learned separately and combined in the classifier phase [41, 42]. Deep learning is an emerging paradigm which has revolutionised cognitive tasks like object recognition, detection, audio and NLP. In this section we review the relevant methods for this thesis.

### 2.2.1 Image Classification

Convnets (CNNs) have been applied to image classification before deep learning became popular [43–46]. However given large scale datasets CNN methods achieve the best accuracy [34, 47–49] due to joint feature and classifer optimisation. This was firmly established by Alex Krizhevsky *et al.* [34] who developed the AlexNet architecture achieving the best performance in ILSVRC 2012. Since then other architectures have been used to progressively make the accuracy better by increasing

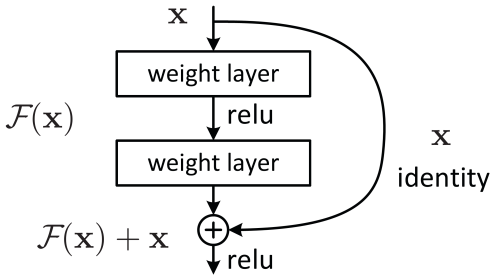



FIGURE 2.1: The ResNet building block (Adapted from [5]). $x$ is the input, $F(x)$ depict the convolution operations and identity connection depicts the skip connection from input to output.

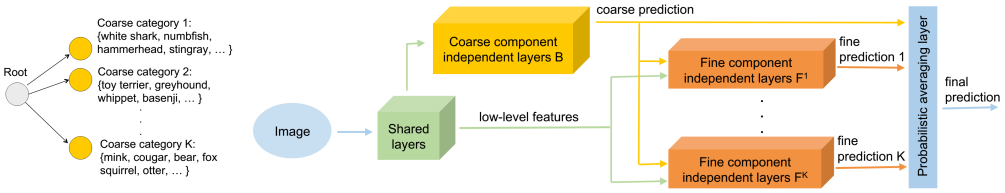



FIGURE 2.2: The HD-CNN architecture depicting the coarse to fine grained classification (Adapted from [56]).

depth [50, 51] or other techniques like the Inception architecture by Google [38]. Recent advances in this direction include the ResNet method [5]. The main idea of the method is to facilitate the gradient optimisation by adding identity skip connections. Figure 2.1 depicts the basic building block of a residual network. By stacking these residual units into very deep networks (¿1000 layers deep) the state-of-the-art performance in image classification, object detection have been achieved [5].

For very large number of classes tree based hierarchical classifiers have been used [52, 53]. The tree is built by putting more general classes a level higher than more specific classes, for example Siberian husky, Labrador retriever etc. all belong to the parent class Dog. This way semantic information on classes can be preserved [54]. A method to learn classes and hierarchies automatically was proposed by [55] by grouping similar classes according to similarities and the network self-organises them. In [56] the CNNs are itself made hierarchical. That way the initial CNNs can learn to generalise among the broad categorical classes while the downstream CNNs specialises on the fine grained subcategories. Figure 2.2 depicts this process.

With the recent advent of specialised datasets for Dogs [57], Cars [58], Flowers [59], Birds [60, 61]etc. fine grained subcategory classification has grown in importance. Branson *et al.* [62] have also cast the CNN in a deformable part model framework where different parts of the objects are fed through the network to extract features and then are aggregated. In a similar vein Zhang *et al.* [63] proposed a part-based R-CNN object part detection and grouping. Like the original R-CNN approach selective search [64] is used to generate the part proposals.

### 2.2.2 Object Pose estimation

It should be noted that object classification and pose estimation are orthogonal problems. In this context orthogonal means that the optimisation targets are not compatible. This is due to the fact that a generic detector tries to optimise for pose invariance while the pose classification requires that information to be present in the final layers. In object recognition, ability to recognize an object irrespective of pose or view point is considered a virtue of the classifier. Hence the deep neural network features for object detection or recognition are invariant to pose. Pose also poses a challenging problem because the pose space is continuous. For that reason in tasks like head pose estimation the pose angle is binned into coarse classes [10]. Similarly features for a human body detector and pose estimator are orthogonal. Hence joint optimisation using multi-task learning makes the problem space grow linearly with every additional task. However, since CNNs have very high capacity, these problems have been successfully addressed for example in DeepPose [65] which attains state-of-the-art performance in articulated human pose datasets like FLIC [66] and LSP [67] and significantly outperformed previous state-of-the-art methods based on deformable part models that built a graphical model of parts [68–70].

It has also been shown that although the pooling layers promote, translational invariance in the features, fully convolutional neural networks retain position information in the features. Tompson *et al.* [71] and Chen *et al.* [72, 73] exploited this to learn the heatmap of body part priors and used a probabilistic graphical model to group them into grop them into individuals. This is a bottom up approach compared to the top down approach of detecting humans and then classifying poses.

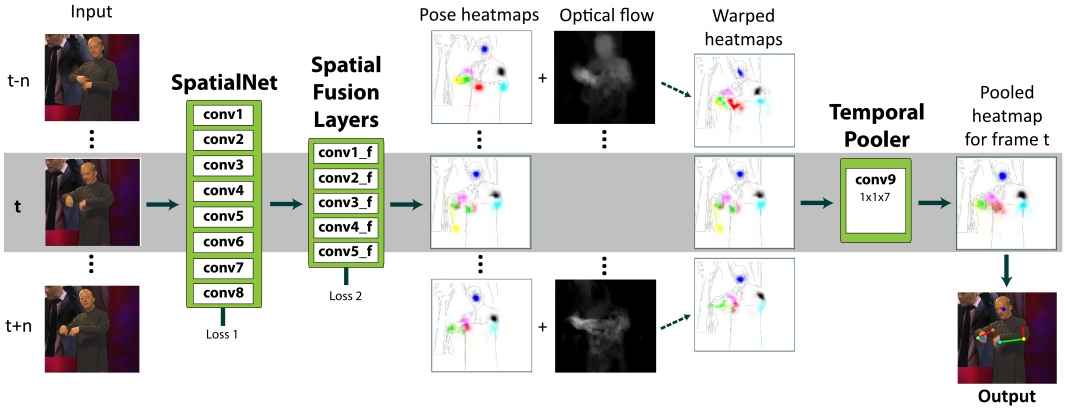



FIGURE 2.3: The Flowing ConvNet architecture used in [74] uses the optical flow and temporal information to stabilise pose estimation (Adapted from [74]).

Pfister *et al.* in [74] exploited the temporal information in videos to propose the state of the art method for pose estimation in videos. They warped the output of the pose model based on temporal optical flow estimation. This introduces a form of smoothing that stabilises the pose estimation. Figure 2.3 depicts the method described in [74].

### 2.2.3 Object Detection

Object detection can be seen as generalisation of object recognition. This due to the fact that object detection can be seen as densely applying object classification neural network across all the image pixels followed by non-maximal suppression. However in practice this can be very expensive computationally to be practical. The first pioneering architecture that used CNNs for object detection came from Girshick *et al.* [75] which used selective search and object proposals to reduce the number of boxes to apply the CNN to. The final features of those boxes were then classified by SVMs and the fine grained bounding boxes were also predicted. In practice this achieved the state-of-the-art performance while reducing compute by a factor of $10^3$. However it was still very compute intensive as the costly neural network had to be applied to a couple of thousand objects predicted by the object proposal algorithm. The biggest flaw was, the object proposal algorithm in the front was decoupled from the neural network. Hence any error of the algorithm was propagated down the computational chain.

To overcome these problems, a Region of Interest (*ROI*) pooling layer was proposed and inserted between the convolutional layers and fully connected (*fc*) layers [76–78]. Around this time fully convolutional architectures were becoming the norm in object recognition like Residual Nets (ResNets) [5] and GoogLeNets [33, 79]. This, we argue is at odds with the objectives of object detection. These networks by design (deep and convolutional) promote translation invariance through successive convolution and pooling operations that is very much desirable for object recognition. However, in turn this leads to the later convolutional layers being less sensitive to the position of the objects. On one hand it might seem prudent to use these networks and apply them fully convolutionally to a large image produce an object based classification heatmap. However this proved to be empirically inferior. Hence in [5] ROI-Pooling layers are inserted between two convolutional layers to break the translation invariance at the cost of more computation.

To solve these problems, methods that embedded the object proposal as a separate but connected CNN branch in the pipeline like the Faster-RCNN architecture [78] was proposed. This enables the two networks to be jointly optimised. This RPN network is trained to class agnostic.

#### 2.2.3.1 Head Detection

Human head detection when compared to face detection, is often a less studied subject in literature. In the modern deep learning framework that requires a considerable volume of training data, has two main methods that have been proposed. Head detection is challenging because, unlike face, heads have a larger degree of variation in appearance and occlusion like hairstyles, head wear etc. The first approach modifies the R-CNN approach with graphical models to jointly detect heads based on their relation to the scene [3]. This approach inherently learns the appearance models of heads and their relations to each other in a scene explicitly. They introduce a large scale Hollywood heads dataset. However, we argue that this assumption is only valid for structured scenes like movies where actors have their spatial relations defined within constraints. This is not applicable to large in the wild crowded scenes like shopping malls, or airports.

Another class of detector called the reinspect was proposed in [80] which uses a fully convolutional neural network to encode an image into a feature volume. The

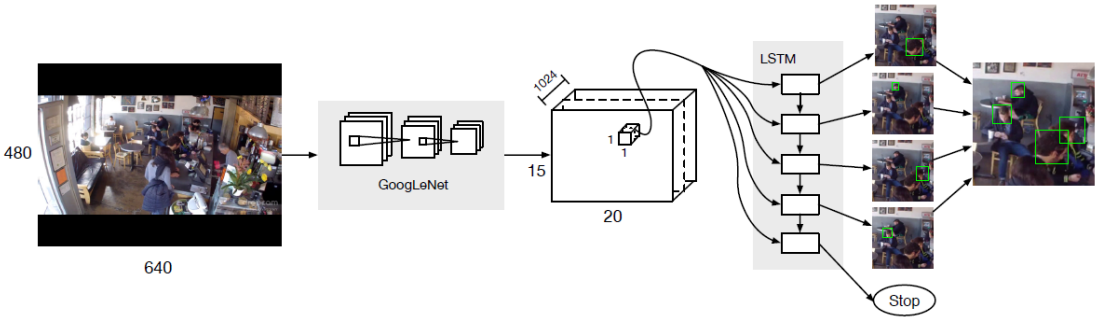



FIGURE 2.4: The Reinspect architecture where a CNN-LSTM architecture is used for human head detection (Adapted from [80]).

last stage is a Long-Short-Term Memory (*LSTM*) based recurrent neural network that decoded the feature volume into detections. The proposed Hungarian loss function promoted the decoding to happen in the order of confidence where the most prominent objects were detected first followed by the less prominent ones in order of confidence. Figure 2.4 shows the proposed architecture.

However since the LSTMs are like fully connected layers, this constrains the network to fixed sized inputs and the performance drops when other images are warped to fit the input size due to distortion in aspect ratio and scale. Hence this architecture is not widely applicable. Furthermore the architecture is capable of detecting only one class. Like we have discussed, detection and pose classification can often be orthogonal. Hence we need to treat end-to-end detection and pose estimation as a multi class detection and classification problem limiting the applicability.

### 2.2.4 Object Tracking

Object tracking has important role in computer vision applications. Tracking allows for incorporating temporal information which often leads to disambiguation. However object tracking and data association is reliant on the robustness of representation. Hence instances of intra class objects like multiple pedestrians, need to have discriminative representation which then must also be discriminative for inter class variation.

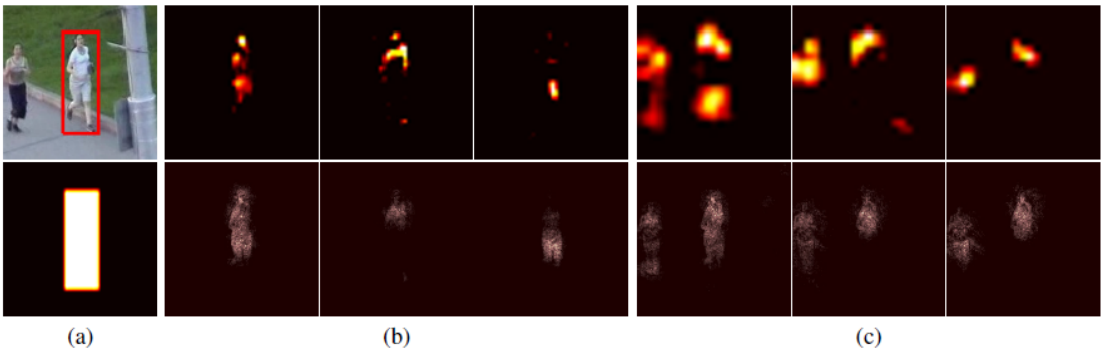



FIGURE 2.5: The CNN features from VGG16 architecture.(a) Shows the input image, (b) Conv 3 features show intra class seperation, (c) Conv 5 features show inter class (category) separation. (Adapted from [81]).

In [82], CNNs features pretrained on imagenet are used for object tracking. The initial detection feature is taken as ground-truth and a heuristic schema is used to update the ground truth object appearance models.

Wag *et al.* [81] propose a multiobject tracker based on the hypothesis that different levels of CNNs have different discriminatory power. For example higher levels have categorical information like object class (pedestrian, bike, plane etc.), but lower level features provide more discrimination among the objects of the same class. Figure 2.5 show that this hypothesis has merit. However, we must note that, in environments like stadiums, where a lot of people only vary subtly in appearance, this may lead to mis-association errors unless we adapt the appearance model on the fly. Also as appearance changes, domain adaptation becomes a necessity.

Another method proposed by Li *et al.* [83] introduces a target-specific CNN for object tracking, where the CNN is adapted to the target with online training. They maintain pool of CNNs where each CNN maintains a specific set of kernels that are updated online instantiated by the initial CNN feature. They use relative low number of filters in the CNN that are updated in the online temporal adaptation.Given the next image, the highest scoring CNN in the pool are used to evaluate the hypothesises and the rest of the models are trained with warm-start back propagation. This approach, though highly adaptable, is very compute intensive. Also it does not explore the design space of the networks itself, and given the high complexity, it is not easy to intuitively see what the optimal architectures might be.

**Pedestrian Tracking**

In pedestrian tracking, typical motion can be learnt by using flow vectors and clustering but often requires a strong assumption that motion patterns are stable [84–87]. Persistent changes can be incorporated over time but ad-hoc trajectories are still typically seen as outliers [88, 89]. The resulting models cannot accurately reflect pedestrian response to spatio-temporal context which could cause tracking failure and data-association errors, particularly if occlusions occur (Fig. 5.2). In such cases an intentional prior (feature) that can predict an ad-hoc change in trajectory is appealing. This theory also generalises to other intentional features: consider a car approaching a crossroads and the indicator light signals intention to turn; contextual knowledge enables better predictions. Several authors have incorporated the concept of 'personal space' and collision avoidance into pedestrian tracking [90, 91]. Others have incorporated the idea that socially grouped pedestrians will attempt to stay in close proximity [92, 93]. Both concepts represent different intentional priors. We propose that head pose as a prior is helpful in predicting model update. This is due to the fact that people tend to look where they are going before deviating from trajectory. No prior work has explored head pose as a prior for tracking.

### 2.2.5 Semantic Segmentation

Semantic segmentation or scene labelling can be viewed as per pixel classification. CNNs, like most other vision problems, have been applied successfully to semantic segmentation as well. Farabet *et al.* [94] first applied CNNs to scene labeling tasks. They input image patches at multiple scales into their ConvNet and show that this approach achieves much better perfromance compared to hand crafted features.The same approach has also been applied to RGB-D semantic segmentation [95].

In [42] an imagenet pretrained CNN model was converted to fully convolutional neural network by changing the fully connected layers to $1 \times 1$ convolutions. Finally a learnable de-convolutional layer was added to upsample to low resolution 32 stride segmentation map to the input dimensions. To increase the resolution, the authors added *jets* of information from lower layers thus increasing the spatial resolution to 8 pixel stride. This approach produced then state-of-the-art results, but introduced new parameters. The resulting segmentations were not very smooth as it did not enforce any spatial smoothness prior.

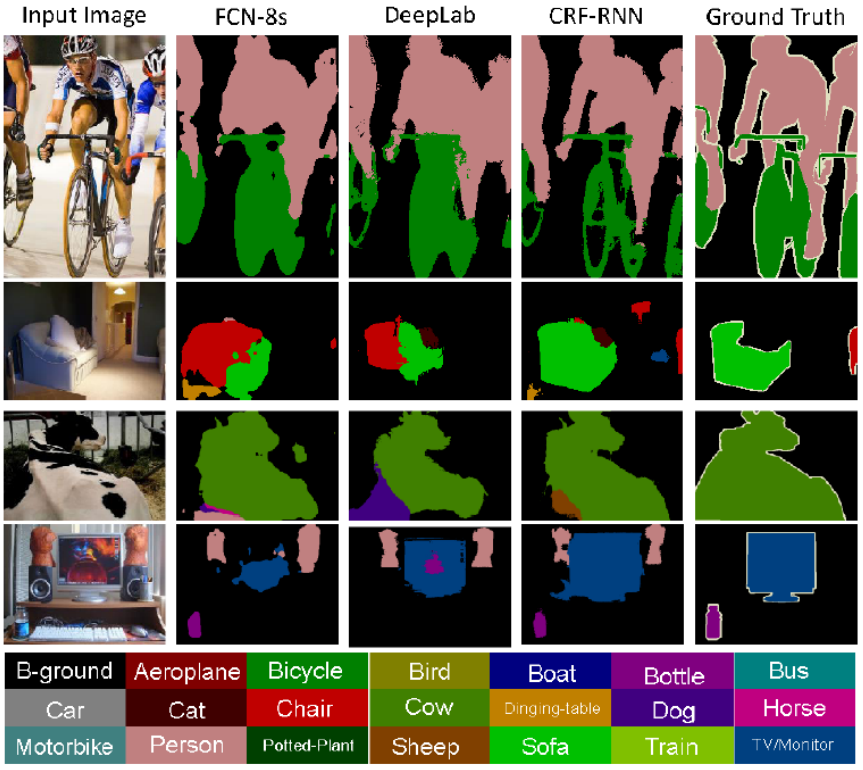



FIGURE 2.6: Illustration of the comparative outputs of the semantic segmentation algorithms on the Pascal VOC dataset(Adapted from [97]).

This was addressed by [96] by introducing densely connected conditional random fields (*DCRF*) to filter the final output. This model achieved a much better segmentation result. However the spatial prior introduced by the DCRF was separate from the CNN and were learned separately while also introducing another parameter as to how many iterations of the DCRF is performed.

This was solved by Zheng *et al.* in [97] by mathematically casting the DCRF inference as an Recurrent Neural Network layer. This allowed for gradient computation and end to end learning of both the CNN and DCRF leading to better performance. Figure 2.6 shows the comparative outputs of the methods on the PASCAL VOC dataset.

Most work since have improved upon these ideas by introducing deeper CNNs like ResNet [5]. Also weak labels like bounding box have been exploited in [98] to increase the amount to training data or conversely reduce the annotation requirement by applying a virtuous cycle of weak labels segmentation to form the ground truth for the next iteration. However we note that none of the works address spare point based labels as is common for many expert annotated datasets.

## 2.3 Discussion

We have identified that there is a gap in the landmark free head-pose estimation research. We aim to contribute a solution when it comes to unconstrained head-pose estimation in mid-low resolution. Briefly, we proceed by evaluating current techniques and augmenting them. Then we also deep learning framework for unified head pose estimation in RGB and/or RGB-D data that spans from high to very low resolutions; we also propose an end-to-end state of the art system that does both detection and pose estimation. We have also found a gap in existing datasets. We solve that by creating a public headpose dataset that fills the major void in head pose research datasets by offering in one unified set, desirable properties in terms of modalities (RGB and Depth), constraints (all poses, not only frontal), quality (accurately labelled for regression) and at the same time one that spans from close to long range resulting in high to low quality data respectively. We also propose to model human gaze and its spatial uncertainty from head-pose as a spherical Von-Misses Fisher distribution on a spherical manifold in $\mathbb{R}^{\mathbf{3}}$ to define person-person and person-scene interaction metrics and evaluating them on comprehensive open datasets. Finally we also apply the robust headpose estimation to address the gap of behavioural tracking with a new tracker called Intentional Kalman Filter.

# Chapter 3

# Feature Design Based Approaches to Head-Pose Estimation

*In this chapter we perform experiments to design features discriminative for headpose estimation in RGB-D data. We evaluate various classification and regression techniques like Support Vector Machines and Gaussian Process Regression. We propose a novel depth normal based feature called Histogram of Depth Surface Curvatures (HDSC). We also evaluate the relative merits of the features and their ideal fusion. We perform temporal smoothing via Particle Filtering. We also establish state-of-the-art results sans deep learning techniques which we investigate in the next chapter.*

In domains where close level iris/eye tracking is not possible, human head pose is the most important factor in determining focus of attention. Head pose estimation has been studied in two distinct domains namely, visual surveillance [4, 99–101] and Human Computer Interactions(HCI) [1, 27, 102] with different methodologies due to difference in the quality of the input. In the surveillance domain the nature of the problem allows the exploitation of priors like walking direction and crowd based behavioural priors to augment the low resolution visual features; while in the close range domain of human computer interaction, more fine grained feature extraction/ Facial Landmark detection approaches can be employed for better accuracy. However in HCI domain the problem has been formulated with natural user interaction in mind, i.e. , the user is always facing(near frontal) the sensor and is fairly close by (not more than 1-2 meters). So the facial landmark based techniques typical of HCI are not applicable to unconstrained head pose estimation at a distance. Furthermore in most indoor interaction scenarios, the subjects are static and there can be a lot of occlusions. Hence the priors like motion direction, body direction etc that are more easily exploitable in a surveillance scenario may not be applicable. Our focus in this chapter is to introduce a system that addresses these issues by estimating unconstrained head-pose in a static scenario even when the subjects are more than four meters away from the camera.

The pioneering work on low resolution head pose estimation was proposed by Robertson and Reid [9] which used a detector based on template training to classify head poses in 8 directional bins. This approach is heavily reliant on skin colour detection. Subsequently this template based technique was extended to a colour invariant technique by Benfold et al. [4]. They proposed a randomized fern classifier for hair face segmentation for the template matching. This work was later improved upon by Siriteerakul et al. in [20] using pair-wise local intensity and colour differences. However like all template based techniques in head pose estimation, these suffer from two major problems: first, it is non-trivial to localize the head in low resolution images; second, different poses of the same person may appear more similar compared to same head pose of different persons.

This has led to the proposals of representing head images in a different feature space that has more discriminatory property for head pose independent of persons. Few non-linear regression approaches like Artificial Neural Networks [103, 104] and High-dimensional manifold based approaches [105, 106] try to estimate the head poses in a continuous range. Chen and Odobez [11] proposed the state-of-the-art

method for unconstrained coupled head pose and body pose estimation in surveillance videos. They used multi-level HOG for the head and body pose features and extracted a feature vector for adaptive classification using high dimensional kernel space methods. These techniques are quite general and don't depend on the heads being in near frontal poses unlike the HCI techniques. Nevertheless high degree of error or uncertainties that arise from these methods, render them unsuitable for tasks like fine grained human interaction or attention modelling. On the other hand, the HCI side of the problem formulation is limited to 2m distance from the sensor along with near-frontal headposes.In [102] an Iterative closest point (ICP) based mesh fitting approach is employed for Head pose detection. In [27] the candidate head poses are rendered and matched to the input depth image and the 6 DOF pose is solved by optimizing via Particle Swarm Optimisation. Fanelli et al.[1] used a randomized patch based decision forest regression for head pose regression. Work on head pose regression for scene and human interaction understanding has been presented in [107]. This work focuses on head-pose regression and interaction detection in 2D movie/ tv-series scenes. While it is quite robust, this only works with yaw angles of $\pm 90^{\circ}$. However this method is quite generalized and does not depend on other priors or specific facial landmark detections.

We have identified that there is a gap in landmark free head-pose estimation research when it comes to unconstrained head-pose estimation in mid-low resolution which we plan to address in this chapter.

The scientific contributions of this chapter are as follows
(a) Definition of a discriminative depth feature called Histogram of Depth Surface Curvature (HDSC) on noisy depth data based on second-order implicit filtering [108]
(b) Detection of unconstrained head-pose azimuthal and pitch angles on a continuous manifold without any other priors.
(c) Definition and validation of a domain specific Gaussian Process Regressor (GPR) [109] covariance function to fit the problem domain , and
(d) Adaptation of particle filtering scheme for temporal smoothing that takes into account regressor confidence to smooth the temporally evolving distribution.


FIGURE 3.1: The headpose images from (a) [110] in RGB, and (b) Our RGB-D dataset.

## 3.1 Datasets

In this section we briefly discuss the datasets used in our work. We have collected many publicly available datasets for standard comparison. However, we have also built our own custom RGB-D dataset to address the shortfall of RGB-D headpose estimation datasets.

### 3.1.1 Head Pose Datasets

In standard literature, headpose has has been evaluated mainly on coarse bins due to low resolution datasets in surveillance domain. Whereas, close proximity HCI style data only contain face poses. The only dataset that has a continuous angular annotation is the Oxford Town Centre dataset [10]. However this dataset is quite small with only a few hundred annotations. Similarly the Caviar shopping centre dataset also has headpose annotation a limited number of videos. To maximise the training corpus, we gathered data from multiple sources that had similar underlying distributions. Datasets annotated for unconstrained face recognition, facial landmark detection all have facial data under various poses. The different head pose datasets that we used are the Oxford town centre dataset, the RGB data from Biwi Kinect headpose dataset [111], the Caviar shopping centre dataset, the IIT Head Orientation dataset along with the IDIAP headpose dataset [110] as shown in Figure 3.1(a) . It should be highlighted that the different datasets have different annotations; some of them have real valued ground truths, others have 6-8 classes spanning the 360°. The datasets vary in resolution from very high in the BIWI dataset to very low in the Oxford town centre dataset. Furthermore no one dataset covers both RGB and Depth along with distance and non frontal poses. Hence we created our own dataset to address these shortcomings.

## 3.2 Creation of Custom RGB-D Dataset

To overcome the lack of large scale data with high quality annotation, we gathered our own dataset. The objectives for gathering the data were as follows
(a) The data should be large scale for training deep neural networks.
(b) The data should have both RGB and Depth modalities.
(c) The data should span from very high to very low resolution.
(d) The data should capture annotations at a granularity of 1 degree or less to evaluate against regression approaches.
(e) Should cover all angles and not only frontal poses.

To this end our own dataset captured 46 people (32 males, 14 females) freely moving in-front of a RGB-D Kinect 2 camera with a miniature wireless IMU sensor hidden inside their hair for head pose ground truth. Each person covered

all possible head pose angles in a continuous manifold at a distance varying from 2m-8m from the camera. We gathered approximately 1500 frames per person giving a total of 68126 examples. The dataset was then manually cleaned for errors and stored as point cloud files. The IMU sensor was calibrated for each person by asking them to look straight towards a point and then setting it as the 0 angle in both yaw and pitch. In Figure 3.1(b) we show a few examples from our dataset. Notice, the data has a very high variation in pose and also resolution.

Since we do not annotate gaze directions or facial landmarks, we collected further statistics about the fidelity of the headpose angle vs. where the person is looking by asking all the participants to sit 2 meters from a wall with markers that were highlighted randomly and the person pressed a button when they saw the marker that was annotated. The participants were asked to look around naturally and not forcefully rotate their heads. We calculated the angle of the accurate headpose annotation of the IMU sensor vs. the actual angle on the wall. This gave us the average deviation of minimum irreducible error of headpose based gaze estimation when we do not track the eye. That error turned out to be around 12.3°.

## 3.3 Design of Head Pose Features in RGB and Depth

We extract features from both RGB and depth modalities because they often provide complementary information. Figure 3.2 shows the extracted features.

### 3.3.1 Histogram of Oriented Gradients (HOG)

The current state of the art result in Head and Body pose estimation at-a-distance was presented in [11]. Their main feature was HOG [25] on the detected humans and heads. Similarly in [107] the HOG feature was again used as an input to the GPR. Hence for the RGB data we retain the standard HOG feature extraction. The input $64 \times 64$ RGB head image generates a HOG feature vector of length 1764. Figure 3.2 (b)(i) plots an example of the HOG feature over the head.

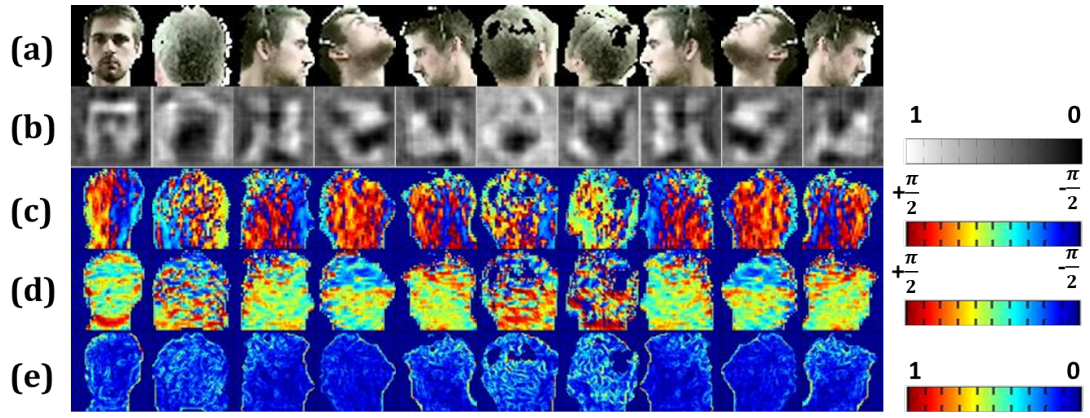



FIGURE 3.2: Here we visualize the various features extracted.Each column shows a different head pose and each row shows a different feature for the corresponding head pose. The heatmap legends have been provided where necessary. (a) This row shows the input Image in RGB (b) This row visualizes the HOG feature through the HOGgles visualizer [112] (c) This row shows the surface normal azimuthal angle as a heat map (d) This row shows the surface normal elevation angle as a heat map (e) The ratio $\Gamma$ as defined in Equation 3.5 is shown in this row as a heatmap

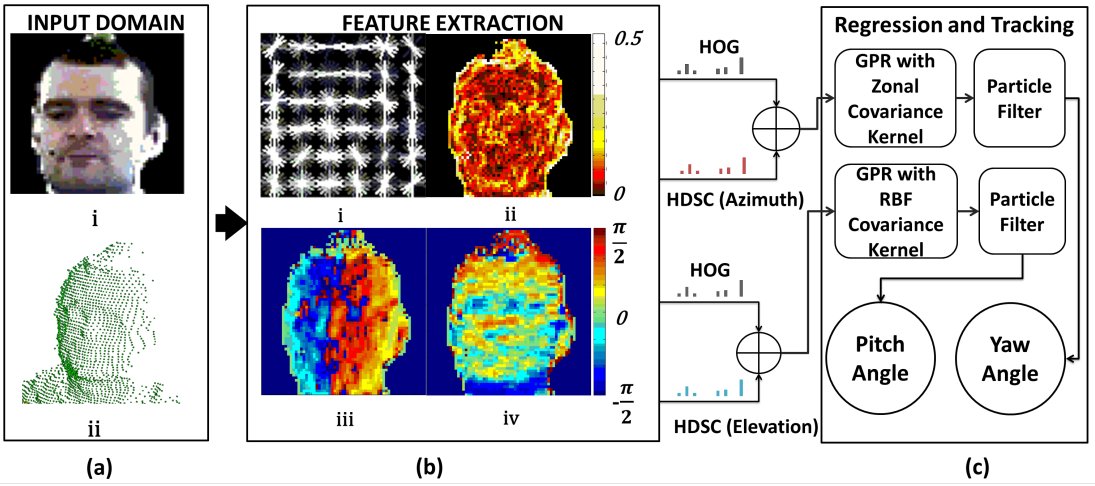



FIGURE 3.3: Here we show the GPR approach for headpose regression. (a) Input modalities, (b)The extracted features *i. HOG, ii. HDSC* , and (c) Regression and tracking

### 3.3.2 Histogram of Depth Surface Curvatures (HDSC)

Another feature which is very informative in depth data is the mapped surface normals of the point clouds. This feature has been very useful object recognition [113]. Apart from surface normals, surface curvatures also encode surface orientation and curvature information. Hence these can be very powerful features in depth maps. Hence we have used the curvatures metric and surface normal

directions to compute a novel feature called the Histogram of Depth Surface Curvatures. (HDSC). To compute the curvature in the organized point cloud field defined by $Z(X, Y)$, which is the value of the depth at depth map co-ordinate $X, Y$, we used the following equations

$$\overrightarrow{N_{X_i,Y_j}} = \frac{\overrightarrow{\partial_X Z_{ij}} \times \overrightarrow{\partial_Y Z_{ij}}}{\left\| \overrightarrow{\partial_X Z_{ij}} \times \overrightarrow{\partial_Y Z_{ij}} \right\|} \tag{3.1}$$

where $\overrightarrow{N_{X_i,Y_j}}$ is the normalized normal vector at $X_i, Y_j, Z_{ij}$ which in turn are the real world coordinate at depth image point $U_i, V_j$and $\overrightarrow{\partial_X Z_{ij}}$ is the X derivative and $\overrightarrow{\partial_Y Z_{ij}}$the Y derivative at point $X_i, Y_j$.

Once we have computed the normals it is easy to compute the curvature tensor by defining the first and second fundamental forms of the local surface patch. Since we are dealing with an 2D organised point cloud, the fundamental forms can be defined in a straightforward manner using the first and second derivatives in the depth image coordinates u,v. We are dealing with data from Kinect like sensors, which present the depth data like a 2D image. As defined above, the depth map is defined over a 2D image grid. Each pixel in this grid has a depth value (Z), a real world X coordinate (X), and a real world Y coordinate (Y) in meters. This form of a point-cloud is called an organised point cloud. We index the depth map grid with the variables $u, v$ to distinguish between real world coordinates and image coordinates. The first and second fundamental forms are computed as follows

$$E = \partial_u Z_{ij} \cdot \partial_u Z_{ij},\ F = \partial_u Z_{ij} \cdot \partial_v Z_{ij},\ G = \partial_v Z_{ij} \cdot \partial_v Z_{ij}, \tag{3.2}$$

$$\begin{aligned} A_{ij} &= -\partial_u N_{ij} \cdot \partial_u Z_{ij} \qquad & B_{1ij} &= -\partial_u N_{ij} \cdot \partial_v Z_{ij} \\ B_{2ij} &= -\partial_v N_{ij} \cdot \partial_u Z_{ij} \qquad & C_{ij} &= -\partial_v N_{ij} \cdot \partial_v Z_{ij} \end{aligned} \tag{3.3}$$

where $\partial_u$ and $\partial_v$ are discrete partial derivatives with respect to the depth map grid.This leads to the definition of the Weingarten curvature tensor $W$ as follows (for notational simplicity we make the subscript i, and j implicit)

$$W = \begin{bmatrix} \frac{AG - B_1 F}{EG - F^2} & \frac{B_2 G - CF}{EG - F^2} \\ \\ \frac{B_1 E - AF}{EG - F^2} & \frac{CE - B_2 F}{EG - F^2} \end{bmatrix} \tag{3.4}$$

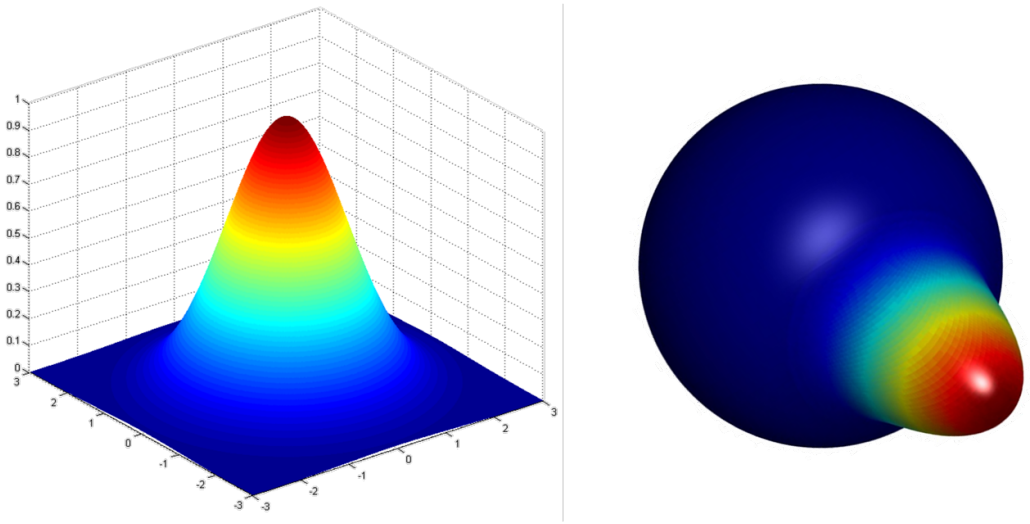
The eigenvectors of $W$, $w_1 = \begin{bmatrix} w_{11} & w_{12} \end{bmatrix}^T$, *and* $w_2 = \begin{bmatrix} w_{21} & w_{22} \end{bmatrix}^T$ give the principal curvature directions and the corresponding eigenvalues $\kappa_1$, $\kappa_2$ are the scalar principal curvature values.

To accumulate the HDSC feature we first compute the following metrics

$$\Gamma = \frac{\kappa_1 \kappa_2}{\frac{1}{2}(\kappa_1 + \kappa_2)}, \ \psi = \arctan\left(\frac{N_z}{N_x}\right), \ \chi = \arctan\left(\frac{N_y}{N_z}\right) \tag{3.5}$$

where $\Gamma$ is the ratio between gaussian curvature $\kappa_1\kappa_2$ and average curvature $\frac{1}{2}(\kappa_1 + \kappa_2)$ , $N_x, N_y, N_z$ are the components of the Normal vector, $\psi$ is the Azimuthal direction of the normal and $\chi$ is the elevation direction of the Normal. In case of some anisotropic patches containing saddle points where $\kappa_1 \approx -\kappa_2$ we set $\Gamma = -1$ . We intuitively define $\Gamma$ to have a positive value at extremums and negative values at saddle points. This definition is more suitable as it implicitly encodes the various facial keypoints. From the aforementioned definitions we can now compute the HDSC feature in local non-overlapping $8 \times 8$ cells. This is done in a similar way to the original HOG feature [25].We compute two histograms one for $\psi$and $\chi$ each. The angles are binned into 9 bin histograms spanning from 0°-180°. For the bin value contribution we use the $\Gamma$. Each neighbouring point contributes to the histograms using a bilinear-interpolation. It can be noted that since depth data is illumination independent, we do not perform any overlapping block normalization. It is noteworthy that the values of the histogram bins can be negative since $\Gamma$ can be negative. Since in surface maxima and minimas $\Gamma$ is always positive and at other points it may be negative or positive, we observe that the parameter $\Gamma$ captures a lot more important information about the surface than the mean-square curvature. Hence the choice of the magnitude metric $\Gamma$ is well justified. Furthermore, we note that our HDSC feature encoding is based upon both curvature $\Gamma$ and normal directions $\psi, \chi$. This definition captures both curvature and normal information into the feature vector, making it more informative than either one on their own.

We have broken down the visualisation of the features to aid human interpretation. In 3.2 (b) the Histogram of Oriented Gradients feature is visualized with the HOGgles tool [112] that tries to reconstruct the information captured by HOG. The HDSC(azimuth) and the HDSC(elevation) features are visualised by visualising the surface normal azimuthal and elevation angles in 3.2(c) and (d) respectively.



**(a)** **(b)**

FIGURE 3.4: (a) The Radial Basis Function Kernel is defined in Linear space using Euclidean distance (b) The Zonal Kernel is defined on a Wrapped Spherical Hyper-surface using geodesic distance

Finally the $\Gamma$ is visualized in 3.2 (e). We have ignored the negative values to increase the dynamic range of the heat-map visualisation.

The final two feature vectors are created by combining the HOG+HDSC(azimuth) and HOG+HDSC(elevation) as shown in Figure 3.3. These final feature vectors are then used to train two Gaussian process regressors[109].

## 3.4 Gaussian Process Regression for Headpose Estimation

To transform the feature vectors into headpose angles on a continuous manifold we use Gaussian Process Regression(GPR) [109]. The choice of gaussian process regression over Support Vector Regression or other regression techniques is mainly due to two reasons. Firstly, GPRs are less dependent on parameter selection and generalize very well; so GPRs can be trained more easily compared to other techniques. Secondly, GPRs inherently model output variance. This value is very useful in determining the uncertainty spread due to regressor confidence.

Gaussian processes [109] are defined as probability distribution over infinite dimensional function space that maps the input to output $X \rightarrow Y$ using $f(X)$. Using this viewpoint any dataset can be assumed to be a part of a single observation from multivariate distribution. A gaussian process is much less parametric than other supervised learning methods. It is defined fully by a mean function, a covariance function and a prior likelihood function. In practice the mean is often assumed to be zero everywhere. The covariance function $k(X, X')$ on the other hand change the distribution on the function space. Hence by changing the nature of the covariance function we can sample different parts of the function space. Here domain knowledge about the problem comes in handy while choosing the functional form of the covariance function. For the pitch angle which has a linear range of $-45° \rightarrow 90°$ in out data, we use a standard radial basis function (RBF) kernel where euclidean distance would suffice.

$$k(x_m, x_n) = \sigma_f^2 exp\left(-\frac{\gamma}{2}(x_m - x_n)^T (x_m - x_n)\right) + \sigma_n^2 \delta(x_m, x_n) \quad (3.6)$$

where $\sigma_n^2$ is the noise variance and $\sigma_f^2$ is the maximum allowed variance. $\delta$ is the Kronecker delta function. $\Gamma$ is the hyper-parameter to be optimized. Figure 3.4 (a) shows the RBF kernel. In case of the yaw angle, it varies in a wrapped circle. Hence, the standard kernels with linear distance metrics are not good models for this. Hence we need to define the kernel as geodesic distance on the curve of a unit sphere in $\mathbb{R}^3$. Such kernels are called zonal kernels [114] and are different from the RBF kernels which depend upon euclidean distance instead of geodesic distance on the feature space manifold. The zonal kernel can be adapted for the Gaussian process regression as follows

$$k(\vec{\mathbf{x}_m}, \vec{\mathbf{x}_n}) = \sigma_f^2 exp\left(-2\varepsilon\left(1 - \vec{\mathbf{x}}_m.\vec{\mathbf{x}_n}\right)\right) + \sigma_n^2 \delta(\vec{x_m}, \vec{x_n}) \quad (3.7)$$

where $\varepsilon$ is the shape hyper-parameter. Figure 3.4 (a) shows the Zonal kernel.

## 3.5 Particle Filtering

Now we introduce a modified particle filtering technique to smooth the observations temporally. Particle filters can be used to compute the posterior probability

of a distribution from a sequence of observations with the Markov process assumption. Let the target state of the system be described by the variable $\xi_t$ at time t and $\lambda_t$ is the corresponding observation at time t, then we can define the posterior probability as

$$P\left(\xi_{\mathbf{t}} \mid \lambda_{\mathbf{t}}\right) \propto P\left(\lambda_{\mathbf{t}} \mid \xi_{\mathbf{t}}\right) P\left(\xi_{\mathbf{t}} \mid \lambda_{\mathbf{t-1}}\right) \tag{3.8}$$

where $P\left(\lambda_{\mathbf{t}} \mid \xi_{\mathbf{t}}\right)$ is the likelihood and $P\left(\xi_{\mathbf{t}} \mid \lambda_{\mathbf{t-1}}\right)$ is the prior probability at time t. It may be noted that out observation $\lambda$ corresponds to the regressor outputs and our prior is defined by our attention metric probability characterized by $\mu$ and $\eta$. In the particle filtering framework the probability distribution is sampled by a set of hypothesis or samples $\left\{s_t^1, s_t^2, ...s_t^N\right\}$. Each hypothesis have a corresponding set of weights $\left\{\Pi_t^1, \Pi_t^2, ...\Pi_t^N\right\}$ that are determined by hypothesis evaluation which is the likelihood of the hypothesis given the observation. After that the particle filtering consists of three steps that are repeated for each observation cycle

1. Sampling : Select samples $\left\{s_{t-1}^{\prime 1}, s_{t-1}^{\prime 2}, ...s_{t-1}^{\prime N}\right\}$ in proportion to weight $\left\{\Pi_{t-1}^1, \Pi_{t-1}^2, ...\Pi_{t-1}^N\right\}$ corresponding to sample $\left\{s_{t-1}^1, s_{t-1}^2, ...s_{t-1}^N\right\}$.
2. Propagation : Propagate samples $\left\{s_{t-1}^1, s_{t-1}^2, ...s_{t-1}^N\right\}$ with state transition probability $P\left(\xi_{\mathbf{t}} \mid \xi_{\mathbf{t-1}}\right)$and generate new samples $\left\{s_t^1, s_t^2, ...s_t^N\right\}$.
3. Weight Computation : Compute the new weights $\Pi_t^N \approx P\left(\lambda_{\mathbf{t}} \mid \xi_{\mathbf{t}}\right)$ corresponding to samples $\left\{s_t^1, s_t^2, ...s_t^N\right\}$
4. Repeat 1-3 for each time step

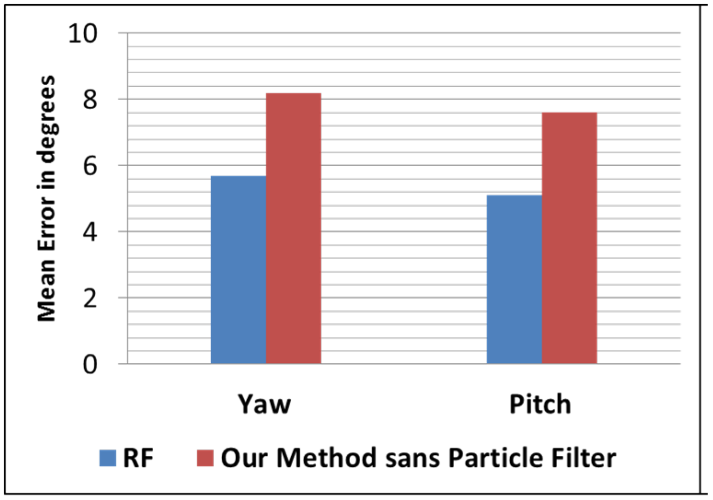
To adapt to our specific problem we define the state transition probability $P\left(\xi_{\mathbf{t}} \mid \xi_{\mathbf{t-1}}\right)$ is defined as follows

$$P\left(\xi_{\mathbf{t}} \mid \xi_{\mathbf{t-1}}\right) = \frac{1}{\sqrt{2\pi}} \exp\left(-\arccos\left(\mu_{\mathbf{t-1}}^{\mathbf{T}} \mu_{\mathbf{t}}\right)\right) \tag{3.9}$$

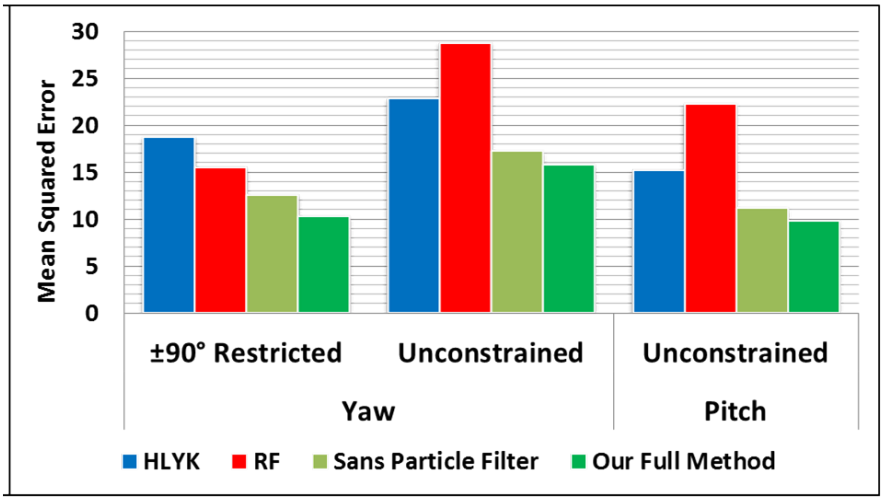
where $\mu_t$ is the output angle of the GPR at time t. Hence the transition probability is the difference between the two angles at time t and time t-1. The likelihood $P\left(\lambda_{\mathbf{t}} \mid \xi_{\mathbf{t}}\right) = \mathcal{L}$ is estimated from the regressor output posterior distribution over output $\mu$ from the training data as follows

$$\mathcal{L} = \int_{\mu'} p\left(\lambda \mid \mu'\right) p\left(\mu' \mid \mathbf{I}\right) p\left(\mu'\right) \mathbf{d}\mu' \tag{3.10}$$

where $p(\lambda \mid \mu')$ is the distribution of regressor outputting $\lambda$ given ground truth angle $\mu'$, $p(\mu' \mid \mathbf{I})$ is the probabilty of the GP output of $\mu'$ given the image $\mathbf{I}$ , and $p(\mu')$ is the prior distribution of the training ground truth data.




(a)




(b)

FIGURE 3.5: (a) Shows the comparative results of [1] and our method without particle filtering on the Biwi Kinect Headpose data. We report the Mean-Error instead of the mean-squared error(MSE) on this data because the same metric was used in [1]. (b) Shows the mean squared errors of the four techniques. We compare the result of the yaw MSE with the head poses limited to ±90°; then we compare the yaw angle MSE on the full dataset; finally we compare the pitch angle MSE

## 3.6 Training and Validation

We validated our method on the BIWI Dataset and our own custom dataset. For training the GPR we split our dataset 50:50. We randomly chose half the samples for training. We extracted the features from these samples and trained the GPR for yaw and pitch angles. We kept the training data fixed for all experiments. Here we present the validation of our technique on two datasets.

## 3.7 Validation on Biwi Dataset [1]

We validated our approach on the Biwi Kinect dataset. The data here is captured very close to the sensor and does not contain back poses. This data lets us compare our general technique to a finer grained HCI technique as presented in [1]. Since the dataset is not always continuous we did not apply the particle filter. Hence the comparison is between their trained random forest and our per frame GPR. This dataset is for validation of our method only since it resides in a completely different domain and we wanted to compare our performance at close range on just near-frontal poses. Figure 3.5 (a) Shows the corresponding results. Here we use mean angular error as measure since the same metric was used in [1]. Next we compare on accuracy by computing the percentage of test data that had an angular error less than 10° on both yaw and pitch. In this metric we achieved 87.35% accuracy compared to 90.4% reported in [1]. This shows that our generalized low resolution technique is only slightly worse than more granular techniques in high resolution data even without our temporal smoothing.

## 3.8 Validation on Our Dataset

Now we report the results on our dataset. Our two baseline methods are the Here's Looking at You Kid (HLYK henceforth) [107] which uses only the RGB data and the Random forest (RF henceforth) based approach [1] which uses only the depth+normal data. From our method we show two results, one sans particle filtering, and finally one with our full method. In unconstrained cases where RF [1] does not detect a nose, we output both pitch and yaw as 0 to compare with

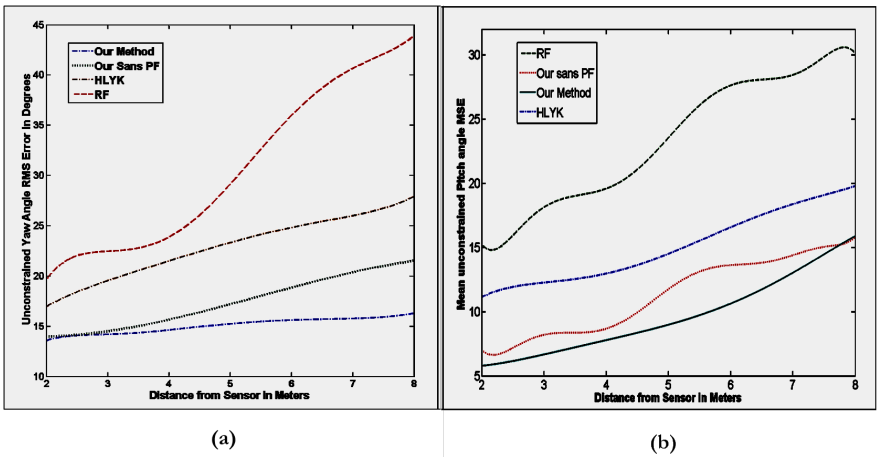



FIGURE 3.6: (a)Plot of the Unconstrained Yaw angle MSE for the different methods with respect to the distance from the sensor, and (b) Plot of the Unconstrained Pitch angle MSE for the different methods with respect to the distance from the sensor

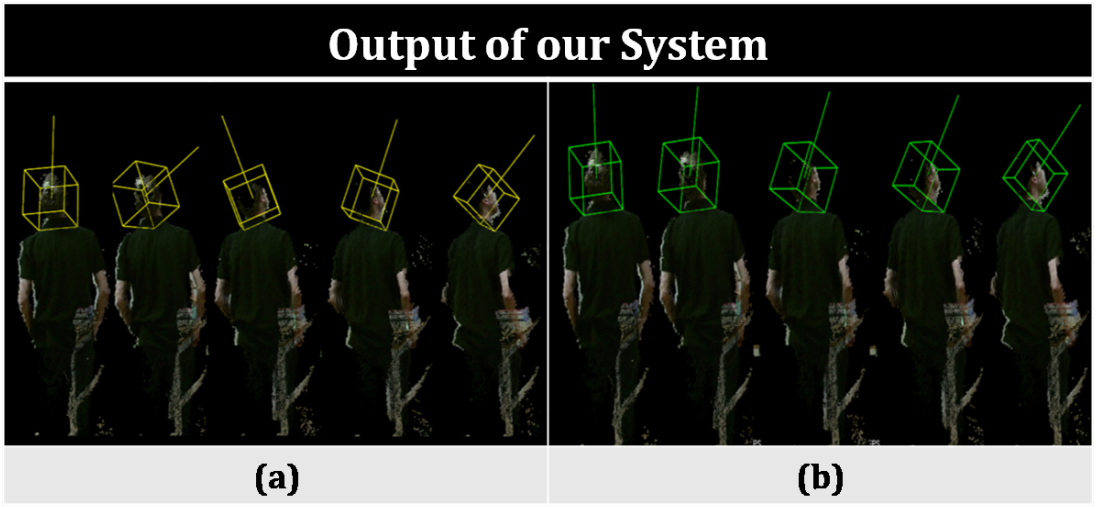



FIGURE 3.7: In this figure we show the output of our system (a) with or (b) without the particle filter. It should be noted that the temporal smoothing provided by the particle filter smooths out the rogue estimation errors due to sensor noise.

the ground truth. This leads to higher error bias for their method hence we have also included the $\pm 90°$ Figure 3.5 (b) shows the results.

Finally in 3.6 (a) and (b) we show the yaw angle and pith angle Mean Square Errors with respect to the distance of the person from the sensor, respectively.

Figure 3.7 illustrates our method under various circumstances.

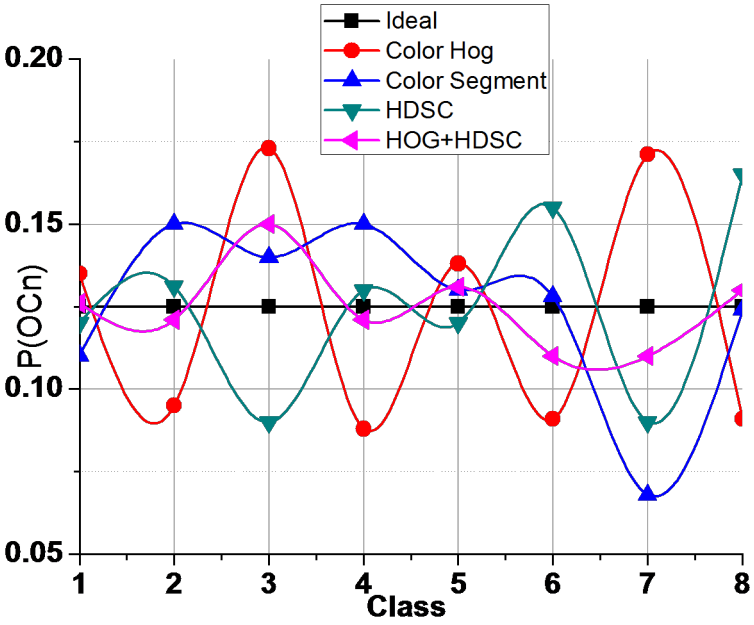



FIGURE 3.8: Probability distribution of Output classes for each type of feature.

## 3.9 Quantitative Analysis of Features

We evaluate the quality of the features with respect to the head angle using a visualisation technique as shown in Figure 3.8. We compare the various features and feature combinations by computing their biases in the 8 headpose bins as shown in Figure 3.2. We first discreetise the ground truth angle and the predicted angle into 45° bins. Then we compute the following metric for each feature in each bin

$$p\left(O_{Cn}\right) = \sum_{i} p\left(O_{Cn}|C_i\right) \times p\left(C_i\right) \tag{3.11}$$

Where,$p\left(O_{Cn}\right)$ is the total probability the classifier outputs class $C_n$, $p\left(O_{Cn}|C_i\right)$ is the conditional probability that the classifier outputs class $C_n$ while the input originally belongs to class $C_i$ and $p\left(C_i\right)$ is the probability of the class $C_i$. In our case where each of the class is represented equally in the test dataset, $p\left(C_i\right)$ is uniform and equal to 0.25. Figure 3.8 shows the probabilistic representational bias of the features for each image class. It is very interesting that the features don't represent information about the classes uniformly. While we see that all the features don't express various poses equally well, we see that our combined HOG+HDSC feature has the least bias.

## 3.10 Discussion

In this section we proposed a novel feature based method for headpose estimation. We also proposed a novel covariance function for the gaussian process and we established the state of the art results on two datasets. However as seen from Figure 3.8 we see that the features have inherent biases. It is suggested that more training data is captured to account for the variations. However, since gaussian processes are inherently sensitive to the number of training samples, because the size of the covariance matrix is based on the number of training samples, and each prediction requires inverting this matrix an operation that costs $O(n^3)$, it soon becomes computationally expensive to introduce more training data to account for more variance in the data. Hence, to make the features more robust by assimilating more training data, we propose to move to deep neural network based approaches that have become very popular recently. In the next chapter we approach the problem with deep learning techniques.

# Chapter 4

# Deep Learning Approaches to Head Pose Estimation

*In this chapter we investigate stronger machine learning and deep learning based methods for the human headpose estimation problem. Previous results show that better features are required for representing human headpose. Although, competitive results were achieved through smoothing, we hypothesize that, the smoothed headpose signal loses information content compared to raw head pose signal and is less useful as a result. To this end, we first build a generative model based representation (based on Deep Belief Network or DBN) and then build a convolutional neural network (CNN) based representation that provide state-of-the-art results on both surveillance RGB and low Resolution RGB-D based datasets. Finally we propose an end to end detection and classification architecture for head detection and headpose estimation with a novel multi-task loss. We show that the end-to-end solution improves the accuracy of headpose estimation while maintaining competitive performance in head detection. The studies in this chapter have been published in the following: in International Conference on Image Processing(ICIP) 2015 [18]; in IEEE Transactions on Multimedia (TMM) 2015 [16]*

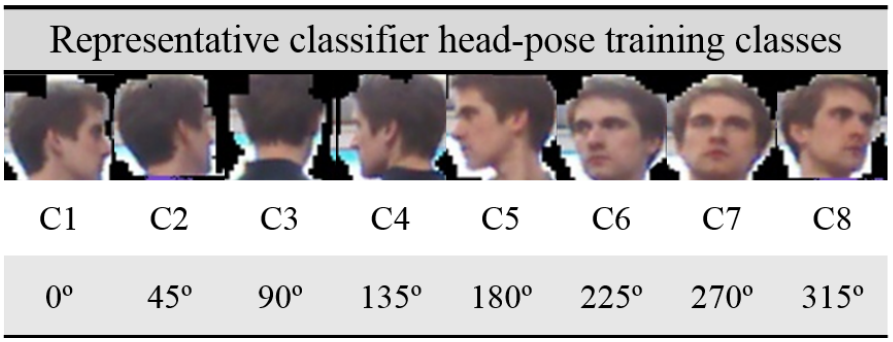



FIGURE 4.1: The headpose class bins as depicted are used in this chapter.

In this chapter, we approach the problem from a deep learning perspective. We first develop a generative semi-supervised approach where we can combine multiple annotated and non-annotated datasets to learn a classification model in Section 4.1. Then we apply CNNs for classification and regression in Section 4.2. Finally we propose a novel DFCNN architecture for end-to-end detection and pose estimation in 4.3. We keep the section self-contained and do the evaluation of each section in that section itself. This is due to the fact that the three approaches solve different problems on different subsets of data. Hence, it is logical to have the experiments and validations contained in the section as well. These three section form the overarching investigation using deep learning methods. We present state-of-the art results in all sections.

## 4.1 A Generative Model

This chapter addresses the need for computing low-resolution gaze estimators without reliance on motion priors to smooth the estimate and presents a demonstrably more robust method using deep learning based generative model. In summary, the main scientific contributions of this section are:
(a) Learning a generative human head model in an abstract head space that can reconstruct heads from low resolution, noisy inputs; (b) Discriminating between head pose angles from the input image without other prior information using multi label discriminative training using various loss functions; (c) We report state-of-the art results on two publicly available datasets when compared to the state-of-the-art approaches. Figure 4.1 illustrates the pose classes for classification on the low resolution surveillance data.

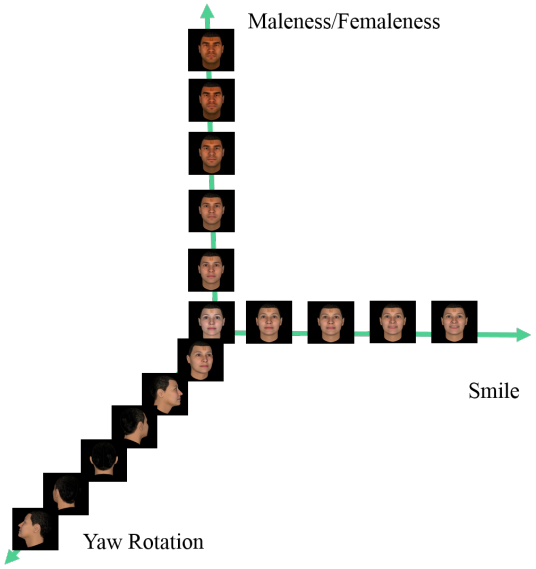



FIGURE 4.2: Conceptual diagram showing different parameters controlling the appearance of the head

### 4.1.1 Parametric Human Head Space

The underlying motivation of this work comes from the theory that human heads lie in a parametric space. This was first shown to be working by Blanz et al. [115] where they derived the basis of this space by a linear combination of shape and texture information of high resolution 3d head scans of 200 adult faces. Hence by using 400 shape and texture parameters they derived a morphable model that could be used to synthesize new faces or estimate a model from 2D images of a given face. However, the human head space is much more complicated because aside from low resolution, surveillance data contains other complicating factors such as varying hair styles, facial hair, and occlusions (e.g. hats, glasses). This requires a much larger parameter space. However, for headpose, which is a very big factor in appearance (and hence has a big eigenvalue in the pca subspace), fewer parameters are needed. This can be understood from the fact that, same person is different poses have a bigger absolute difference in image space compared to disparate people at same angle. Figure 4.2 shows how a parametric head-space can generate various human heads with different identity, expression and pose. The head pose datasets are limited in the number of examples per person and image quality. Hence, we consolidated many different datasets not necessarily ground truthed for head pose into an unsupervised framework in a generative model. Deep Belief Networks [36] are very well suited for this purpose.

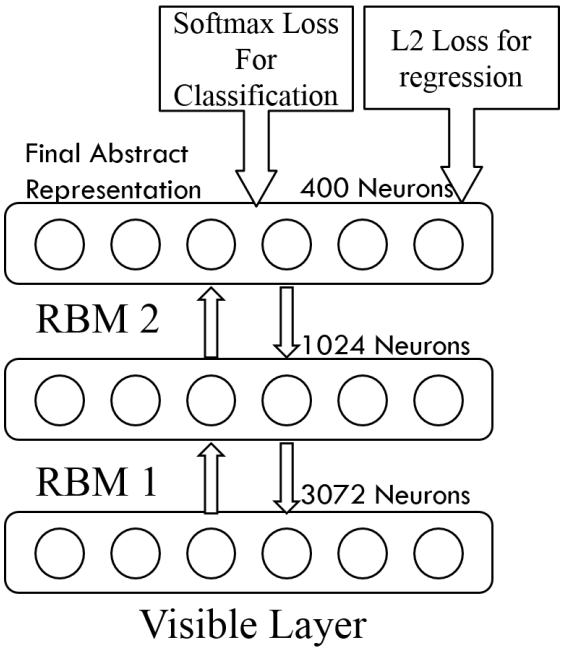



FIGURE 4.3: Here we show the hierarchical DBN architecture which creates a 400 dimensional head representation that is then discriminativly trained on various datasets based with varying ground truths ranging from basic front/back classification to full real valued angle regression with interchangeable softmax and L2 loss functions

### 4.1.2 Deep Belief Networks (DBN)

A DBN is constructed from unsupervised, greedily trained stacks of restricted boltzmann machines (RBMs). RBMs are a form of energy based generative model in which the energy functions can be written as follows:

$$E(v,h) = -b'v - c'h - h'Wv \tag{4.1}$$

Where b,c and W are the parameters $\theta$ and v and h are the visible and hidden units of the model. The model is trained with contrastive divergence that estimates the gradients of the energy function with respect to the model parameters given the training data $\mathbf{X}$.

$$\frac{\partial E(\mathbf{X},\theta)}{\partial \theta} = \frac{\partial \log \mathbf{Z}(\theta)}{\partial \theta} - \left\langle \frac{\partial \log\ f(x,\theta)}{\partial \theta} \right\rangle \tag{4.2}$$

where $\mathbf{Z}(\theta)$ is the partition function defined as

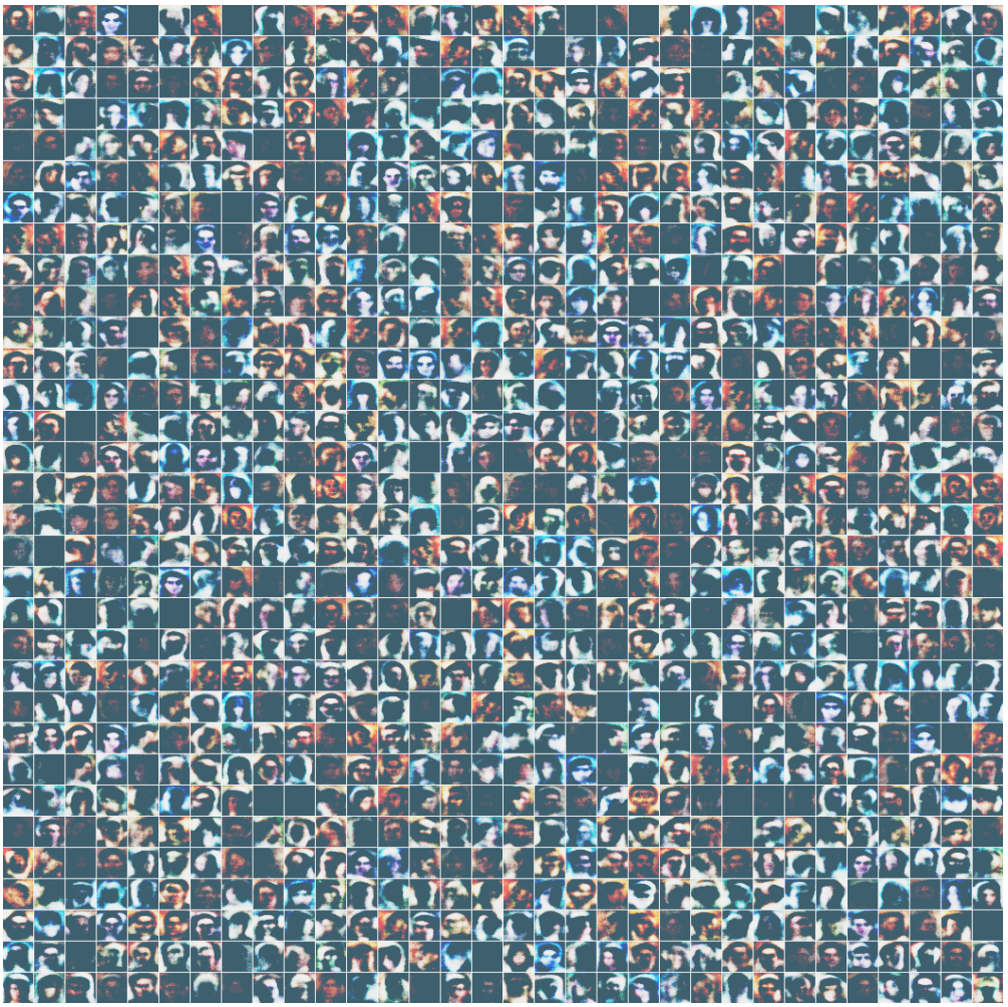


FIGURE 4.4: Visualisation of the RBM weights. Each RBM unit connects to the entire visible input. The weigths are reshaped into visible layer sized squares.

$$\mathbf{Z}(\theta) = \int f(x, \theta)\, dx \tag{4.3}$$

Where $f(x, \theta)$ is the underlying distribution we are trying to model. It is not easy to find the derivative of the partition function because we do not know the underlying representation. It can be suitably derived by using Markov Chain Monte Carlo sampling from the training data and given sufficient examples it should converge to the real derivative, however, this is not computationally tractable. The parameter update equation derived from just one step of Markov Chain Monte Carlo sampling from the training data has empirically proven to be effective by Hinton et al [36]. It can be written as:

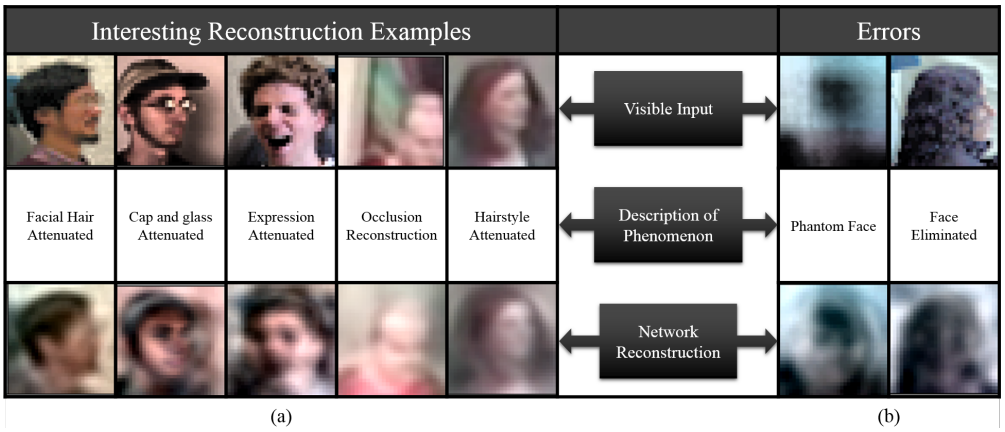



FIGURE 4.5: What the network sees. This figure shows a reconstruction of the input image in top row with their reconstruction from network parameters in the last layer in the bottom row. Sub-figure (a) on the left suggests that for head pose the eye and mouth region is very important whereas facial hair, hairstyle and facial expressions are attenuated. The network has learned to handle occlusions and shift. Sub-figure (b) on the right shows some interesting errors made by the network. Under extreme low resolution or noisy input on the left the network sees a face where none exist. On the right the face is eliminated. However even in these extreme low resolution cases the network can estimate parameters.

$$\theta_{t+1} = \theta_{t+1} + \eta \left( \left\langle \frac{\partial \log f(x,\theta)}{\partial \theta} \right\rangle_{\mathbf{X}^0} - \left\langle \frac{\partial \log f(x,\theta)}{\partial \theta} \right\rangle_{\mathbf{X}^1} \right) \tag{4.4}$$

Where $\eta$ is the training rate. The layers of RBMs are trained in an unsupervised fashion layer by layer to form a Deep Belief Network. Conceptually, by changing the number of neurons in each subsequent hidden layer the representation of the underlying data can be learned in a hierarchical fashion. Figure 4.3 shows the architecture used for our system. We use only two layer because more layers degraded performance with the amount of data present for training. Figure 4.4 shows the learned weights from the first RBM layer.

### 4.1.3 Experiments and Validation

We use multiple datasets to train our system and we validate our approach on two publicly available datasets. Furthermore, for regularisation of the network in the unsupervised phase we included the Multi-task Facial Landmark Dataset (MTFL) [117] and the Labelled Faces in the Wild [118] datasets as they have a wide range of poses, but these are not labelled for head pose. For the unsupervised training

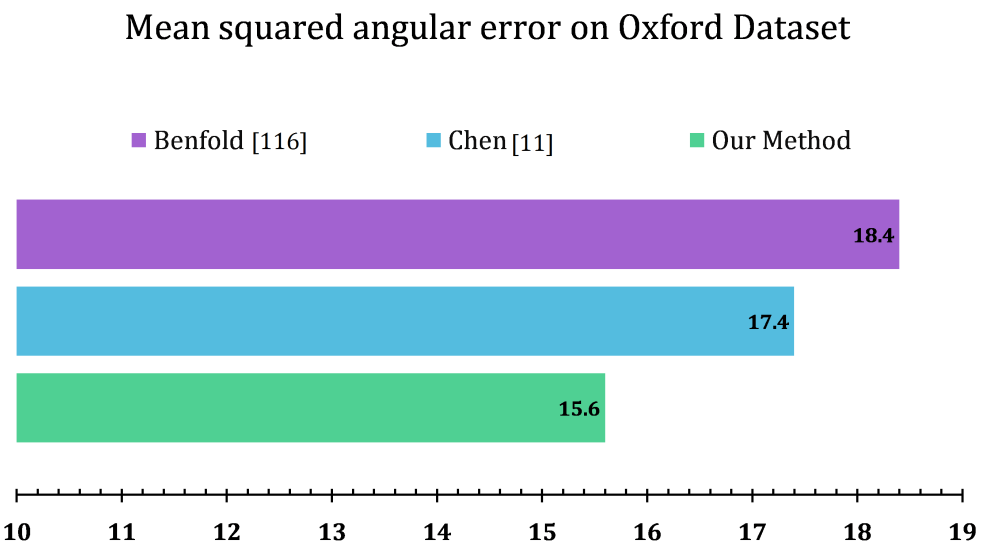



FIGURE 4.6: This graph compares our algorithm in terms of MSE with the Benfold [116] and Cheng algorithm [11]

we use all the available datasets without any labels. For the final fine tuning using labelled data, we randomly selected 50% of the data from our dataset.

### 4.1.4 Training

The network used two RBMs stacked to form a DBN as shown in Figure 4.3. The final output layer was interchanged for various head-pose datasets depending on their ground truth. We normalized all the head images to $32 \times 32$ for input to the network. We also scaled the head bounding box to 0.8, 1, 1.5, 1.8, 2.0, and 2.5 scaled crops to achieve some scale invariance. To achieve translation invariance we also used scale 1 crops with strides of (3,3) pixels from the 1.5-2.5 scaled crops. The network was trained with 30% dropout and a decaying learning rate. For validation on the Caviar and the Oxford datasets we use a training-testing split of 70%-30%. Figure 4.5 shows the reconstructions of the image from the parameters estimated by the networks top most layer by back projection into the image space. This gives us a unique perspective into what the network actually found important for the problem feature selection.

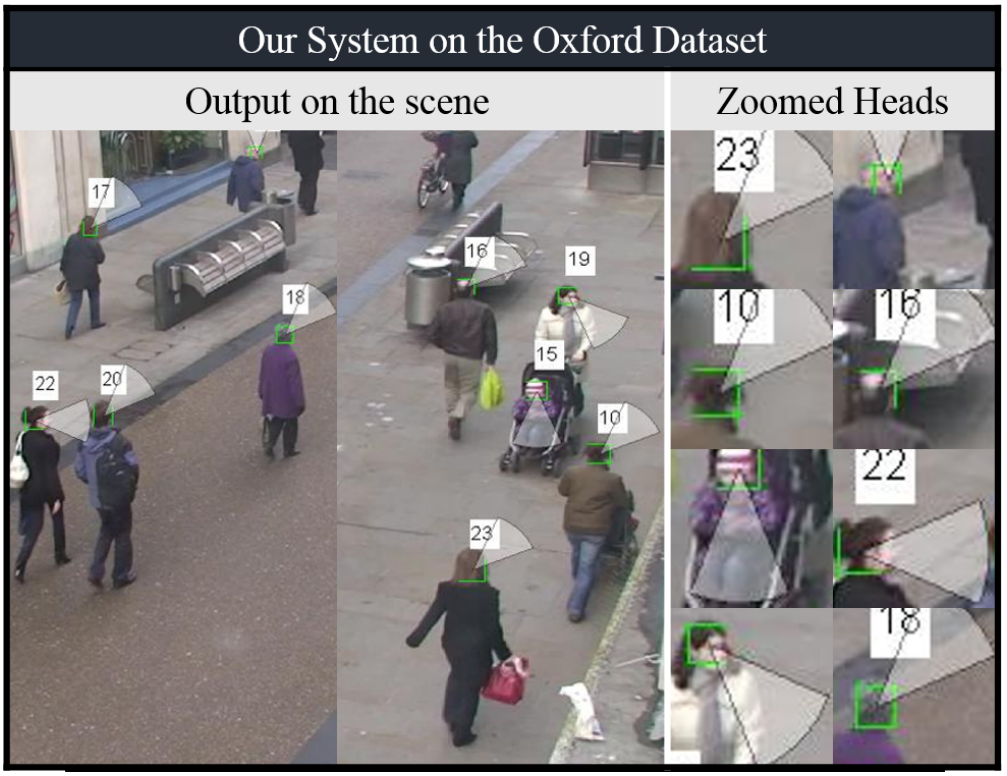



FIGURE 4.7: The output of our method on the Oxford town centre database. The head regions are zoomed in to show the headpose angles.

### 4.1.5 Results

We report our results on the Oxford and the Caviar datasets. In these datasets we classify the head pose into 8 equally spaced (45°) angular bins as shown in Figure 4.1. For comparison with [11] and Benfold [116] we use the Oxford dataset in which both have reported results. One consideration has to be made while comparing because [11] reported the mean square error (MSE) which they derived from a weighted combination of their 8 class classifier output multiplied with the bin angles as $\sum_{i=1}^{8} p_i \overrightarrow{\eta_{\theta_i}}$ where $p_i$ is the classifier output value for the class $i$ and $\overrightarrow{\eta_{\theta_i}}$ is the unit vector in that angular direction. Since our softmax layer gives probability, it is unclear how to interpret vector addition weighted by probability. But for the sake of comparison we derive our mean squared error (MSE) in the same way. Figure 4.6 shows the comparison between our method with the previous state of the art results. In terms of MSE we outperform the best results by 1.8°. The margin while comprehensive may not be representative of the true picture. We therefore present the confusion matrices on the Oxford and Caviar datasets. In terms of classification accuracy on the Caviar dataset we achieve 76.38% accuracy on the Caviar dataset. To our knowledge it is the best result on

Oxford data/Benfold algorithm

| Ground Truth \ Classifier Output | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 0.12 | 0.23 | 0.33 | 0.05 | 0.02 | 0.03 | 0.07 | 0.15 |
| | 0.05 | 0.47 | 0.31 | 0.05 | 0.01 | 0.01 | 0.07 | 0.04 |
| | 0.03 | 0.41 | 0.37 | 0.09 | 0.03 | 0.02 | 0.03 | 0.02 |
| | 0.01 | 0.14 | 0.43 | 0.17 | 0.08 | 0.13 | 0.03 | 0.01 |
| | 0.01 | 0.04 | 0.24 | 0.19 | 0.45 | 0.06 | 0.02 | 0.00 |
| | 0.01 | 0.03 | 0.18 | 0.11 | 0.38 | 0.23 | 0.07 | 0.02 |
| | 0.01 | 0.01 | 0.08 | 0.16 | 0.20 | 0.41 | 0.12 | 0.01 |
| | 0.04 | 0.02 | 0.10 | 0.10 | 0.09 | 0.19 | 0.33 | 0.12 |

(a)

Oxford data/Our algorithm

| Ground Truth \ Classifier Output | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 0.92 | 0.00 | 0.01 | 0.01 | 0.00 | 0.03 | 0.01 | 0.02 |
| | 0.05 | 0.76 | 0.09 | 0.02 | 0.01 | 0.04 | 0.01 | 0.02 |
| | 0.05 | 0.00 | 0.54 | 0.04 | 0.12 | 0.03 | 0.10 | 0.12 |
| | 0.02 | 0.00 | 0.02 | 0.17 | 0.26 | 0.00 | 0.42 | 0.11 |
| | 0.01 | 0.00 | 0.00 | 0.00 | 0.79 | 0.01 | 0.01 | 0.18 |
| | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 | 0.98 | 0.01 | 0.00 |
| | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 |
| | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 |

(b)

Caviar data/Our algorithm

| Ground Truth \ Classifier Output | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 0.90 | 0.01 | 0.02 | 0.01 | 0.01 | 0.00 | 0.01 | 0.03 |
| | 0.08 | 0.50 | 0.27 | 0.04 | 0.02 | 0.00 | 0.04 | 0.05 |
| | 0.01 | 0.01 | 0.91 | 0.01 | 0.02 | 0.01 | 0.02 | 0.01 |
| | 0.02 | 0.02 | 0.42 | 0.25 | 0.17 | 0.05 | 0.07 | 0.00 |
| | 0.00 | 0.00 | 0.03 | 0.01 | 0.91 | 0.03 | 0.03 | 0.00 |
| | 0.00 | 0.00 | 0.01 | 0.00 | 0.03 | 0.89 | 0.05 | 0.01 |
| | 0.00 | 0.00 | 0.01 | 0.00 | 0.01 | 0.02 | 0.95 | 0.01 |
| | 0.00 | 0.01 | 0.03 | 0.00 | 0.01 | 0.01 | 0.09 | 0.80 |

(c)

FIGURE 4.8: Confusion matrices showing the output of (a) The Benfold algorithm [4] on the Oxford town center dataset, (b) Our DBN approach on the Oxford town center dataset, (c) Our DBN output on the Caviar dataset.

this dataset. Figure 4.7 shows the output of the DBN on the Oxford data. On the Oxford dataset, for comparison, we also show the output confusion matrix of the Benfold algorithm [4] along with our confusion matrix. Apart from the fact that we outperform the Benfold algorithm by a large margin, it is interesting to note that the Benfold algorithm shows some interesting biases connected to walking direction. The confusion matrix shows a large classifier bias in the C2 and C6 pose classes, which, as can be seen from Figure 4.1, coincides with the direction of the road. As most people are going up or down the road and generally looking where they are going (as can be seen from Figure 5.1) the algorithm seems to have learned this bias in the scene.

We out perform both the previous state of the art methods without using any kind of prior coupling as the confusion matrices in Figure 4.8 show very clearly. For completeness we show the MSE in Figure 4.6, as this metric is used in the papers against which we compare. The difference in MSE is not as dramatic as the confusion matrices shown suggests but nevertheless demonstrates a significant improvement. One feedforward pass through our DBN on a GPU for headpose estimation on a single $32 \times 32$ image takes 0.8 milliseconds. This makes our system real-time and it can be scale up massively but still maintain real-time performance.

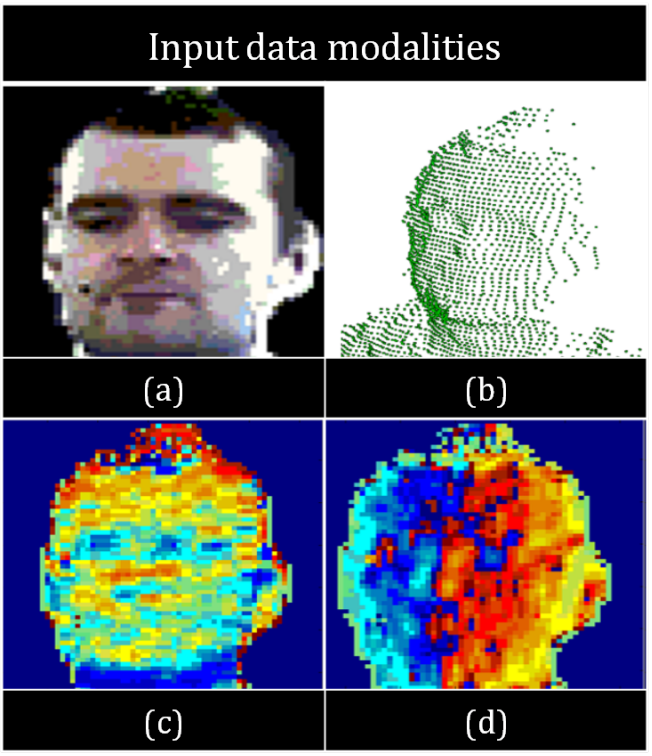



FIGURE 4.9: The input modalities. (a) Shows the RGB input. (b) (c) and (d) Show the Depth, surface normal elevation angle and the surface normal azimuthal angle respectively that form the three channels of the DAE encoding

## 4.2 Convolutional Neural Network for RGB-D

In this section we develop a Multi-Modal Convolutional deep Neural Network based headpose estimation method that works across RGB and depth modalities as well as spans the gamut of low resolution surveillance to high resolution human computer interaction (HCI) domains.

### 4.2.1 Deep learning and Convolutional Neural Networks (CNN)

In this section we do not concern ourselves with the problem of detecting heads. Instead we can adapt the output of any head detector and normalize the heads to $256 \times 256$ as input to our algorithm. Once we have the normalized RGB-D heads as input the rest of the process can be briefly summarized as follows. First, if available, we encode the depth image using a scheme that we name DAE encoding which encodes the depth modality with three channels of depth, surface normal

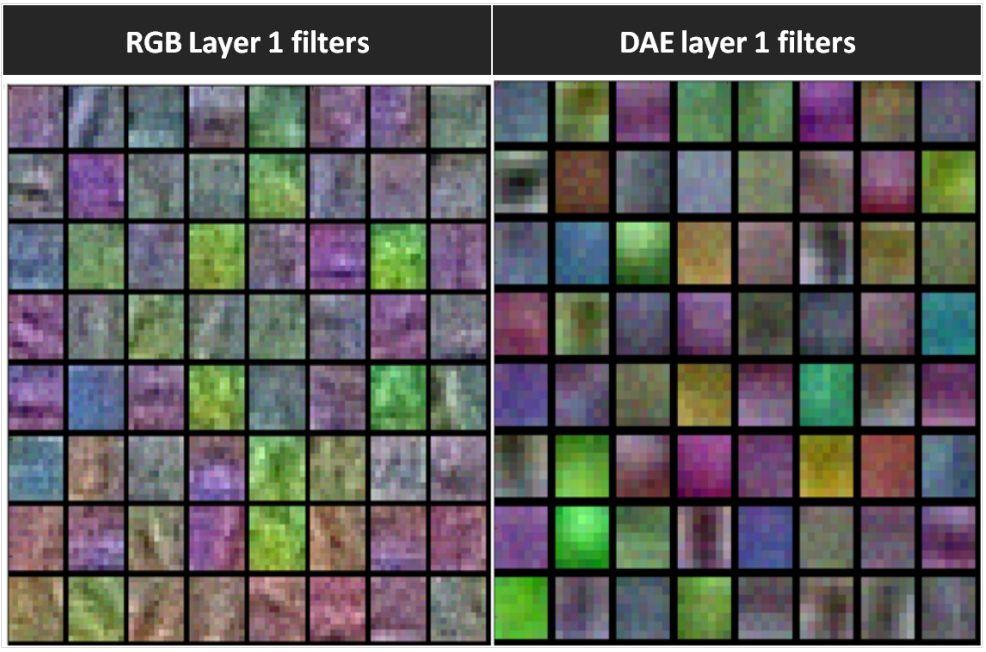



FIGURE 4.10: A visualisation of the first level of learned filters on both the RGB network and the depth network

azimuthal and surface normal elevation angle as shown in Fig. 4.9 and is similar to HHA of [41]. We do not encode the preferred gravity direction in DAE like HHA as all our heads are upright. These inputs are then used to train a CNN each for RGB and DAE and we call them RGB CNN and Depth CNN respectively. The combination of the posteriors of the two CNNs are called the RGB-D CNN.

Convolutional neural networks belong to a class of fully supervised deep models that have proven to be very successful in a wide variety of tasks. The power of CNNs lie in the ability to learn multiple levels of non linear transforms on the input data using labelled examples through gradient descent based optimizations. The basic building blocks of CNNs are fully parametrized (trainable) convolution filter banks that convolve the input to produce feature maps, non-linearities (like sigmoid or Rectified Linear Units/ReLU), pooling layers/downsampling layers (e.g. max pooling, mean pooling etc.) that downsample the feature maps, and fully connected layers. CNNs in particular through their multiple levels of convolution and pooling achieve a high degree of translation invariance in their features. Recent studies from Simonyan and Zisserman [48] have shown that deeper models with smaller filters achieve great expressive power in terms of learning powerful features from data in tasks like object recognition on large scale datasets like the Imagenet [119]. As the model go deeper the number of weights/ parameters or the

networks grow significantly. It then becomes imperative to use large scale labelled training data to train these networks. However one should note that the number of parameters in the convolution layers are orders of magnitude lower than the fully connected layer [120]. Hence by having more convolution layers helps alleviate the problem of this parameter explosion while retaining the expressive properties on the deep models. One such model is the recently introduced Googlenet model [38]. We train two CNNs on the RGB and depth modalities based on this architecture [38]. This architecture has the state-of-the-art results on the Imagenet dataset [119]. In our experiment the same network also gave the best results on our task. The advantage of this network lies in that it is very deep but has a lot less parameters (around 5 million) compared to other contemporary networks like the VGG-16 [48] which has more than 130 million parameters. This lets us train the networks using considerably less training data. We modified the network by changing the Rectified Linear Unit non-linearities (RELU) with Parametric Rectified Linear Unit (PRELU) and their corresponding weight initialisation introduced in [121].

The non-linearities are defined as follows

$$RELU\,(x) = \begin{cases} x & if\ x > 0 \\ 0 & if\ x \leq 0 \end{cases} \tag{4.5}$$

$$PRELU(x) = \begin{cases} x & if\ x > 0 \\ mx & if\ x \leq 0 \end{cases} \tag{4.6}$$

where $m$, the slope in the negative $x$ is a learned free parameter.

The reason the PRELU activations are better than their RELU counterpart lies in the fact that PRELU activations have non zero outputs and non zero gradients in the negative values. This makes them easier to propagate gradients from. Whereas in case of RELU, if some neuron's output becomes less than equal to zero, its gradients also vanish and it hampers learning through gradient descent. The motivation for doing it is that this small change, without increasing the number of parameters of the network significantly improves the accuracy (see [121]).

We also exploit the ability of CNNs to learn from multiple types of labels for the same kind of underlying data to achieve a valid representation learnt on the

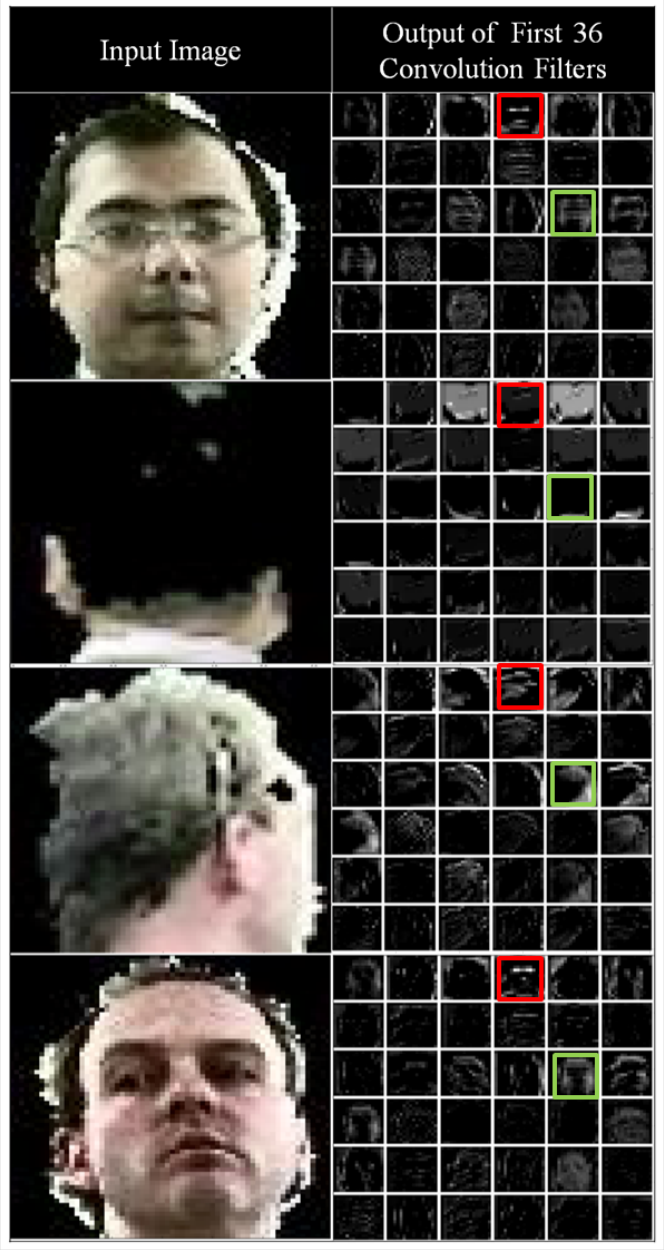



FIGURE 4.11: A visualisation of the features extracted after the first level of convolution. We show only the first 36 channels out of the 64 channels. It is easy to see that some filters are bringing out the facial landmarks (top red where they are detected and bottom red where they are not) whereas others have learned skin maps (indicated in green) among other real facial features.

data. Since there are few explicit head-pose regression datasets, we initialize the training of models with classification into 8 head pose classes spanning 360 degrees. The representative head-pose classes are shown in Fig. 4.12. We learn an initial representation that is then transferred to the regression network and fine tuned for regression. Fig. 4.12 also shows how the CNN features separate easily in only two dimensions whereas the HOG feature that is used in other techniques including

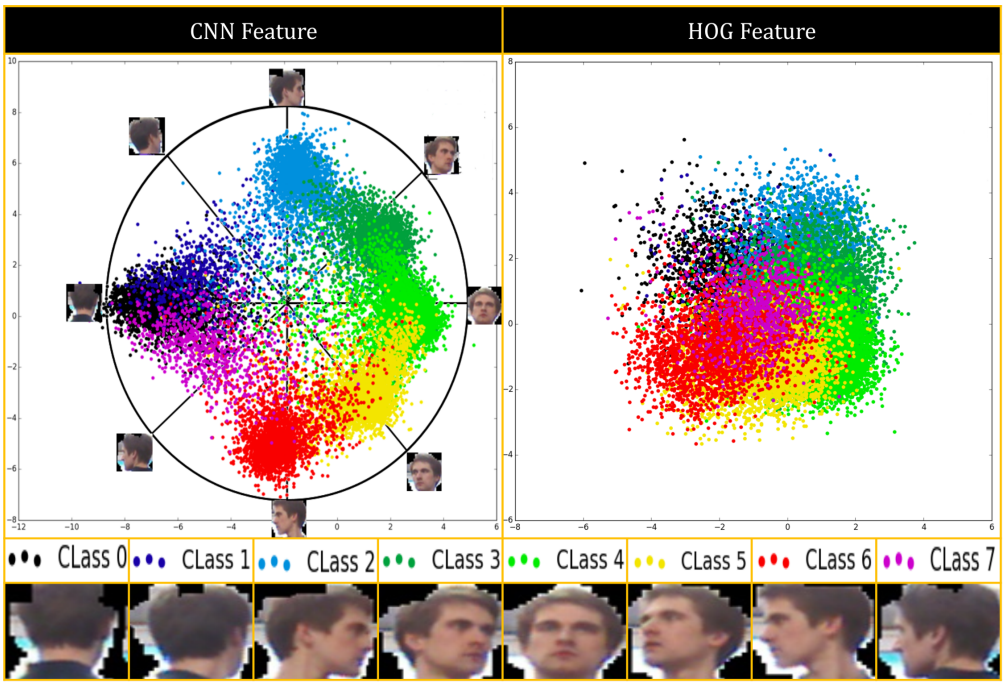



FIGURE 4.12: The Linear Discriminant Analysis (LDA) projected scatter plot of the clusters of the head pose classes after initial training for classification. We compare it with the HOG feature [25] which is most common in the comparable literature. Not only are our clusters well separated, they maintain the approximate closed topology of the circular head-pose manifold. The clusters have their mean near the pose class angles and spread around the circumference of the manifold. This validates our choice of transferring this network to the regression task

[11, 28, 122] is nowhere near as effective. It can also be seen from Fig. 4.10 and 4.11 that the network learns filters, some which could have been developed by intuition, where as other features are not as intuitive but effective nonetheless. It is interesting to note that the feature space embedding presented in Figure 4.12 shows that the CNN learns the implicit circular geometry of the view manifold from the data itself. This is in contrast to [30] where this shape is imposed as a prior assumption. However due to imaging noises and low resolution they might not lie in an ideal circle. Besides, ideal circular distribution may or may not be ideal for a classifier as can be seen from Fig. 4.14. Hence, it is our belief that an end to end approach without prior assumption leads to better results for classification.

For regression we expect to see a similar distribution that is more evenly spread out on the manifold instead of forming clusters. Figure 4.13 shows the output scatter plot of the first two LDA components of our fine-tuned features on regression on our dataset.

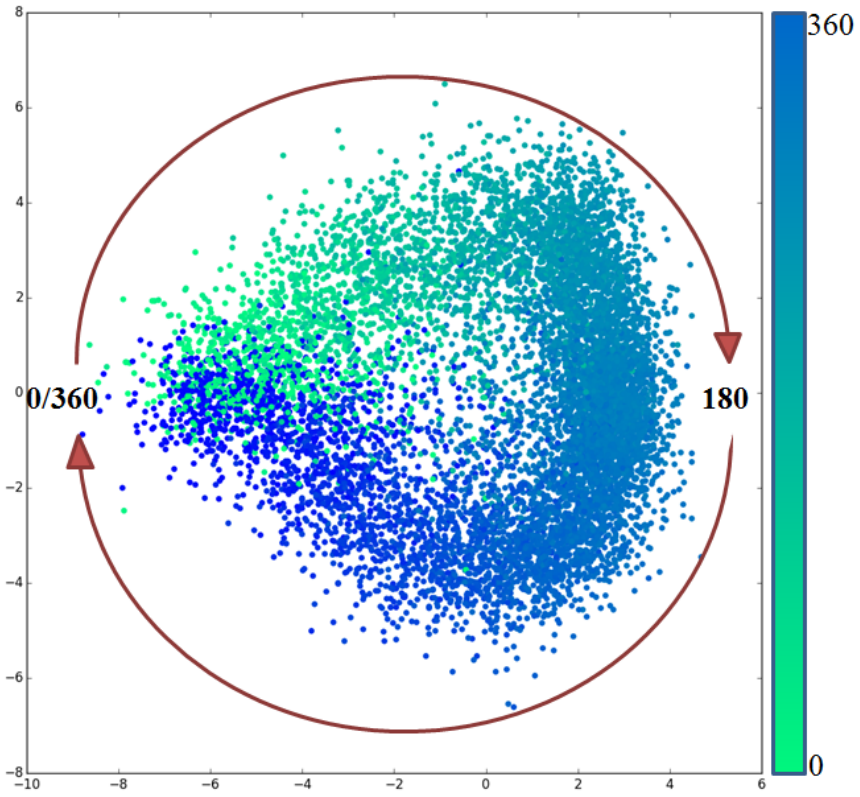



FIGURE 4.13: Linear Discriminant Analysis (LDA) projected scatter plot of regression features on our dataset with a colour map that spans the range 0-360 degrees. The features maintain the manifold.

The regression output is then combined heuristically to obtain a probabilistic attention distribution which we parametrise as a Von-Misses Fisher distribution. This distribution captures two important properties of the head pose regressor output. First, it inherently models the regressor output confidence directly into the distribution concentration parameter $\eta$; second, it also models the inherent irreducible uncertainty in every gaze tracking technique where eye balls are not tracked. We have performed experiments to determine the mean discrepancy between eye and head-pose to model this phenomenon.

We now discuss each of these steps in detail in the subsections that follow.

### 4.2.2 DAE depth encoding

For depth data it is important to encode some spatial and surface information into the data itself, as shown by Gupta et al. [41]. We follow a similar approach

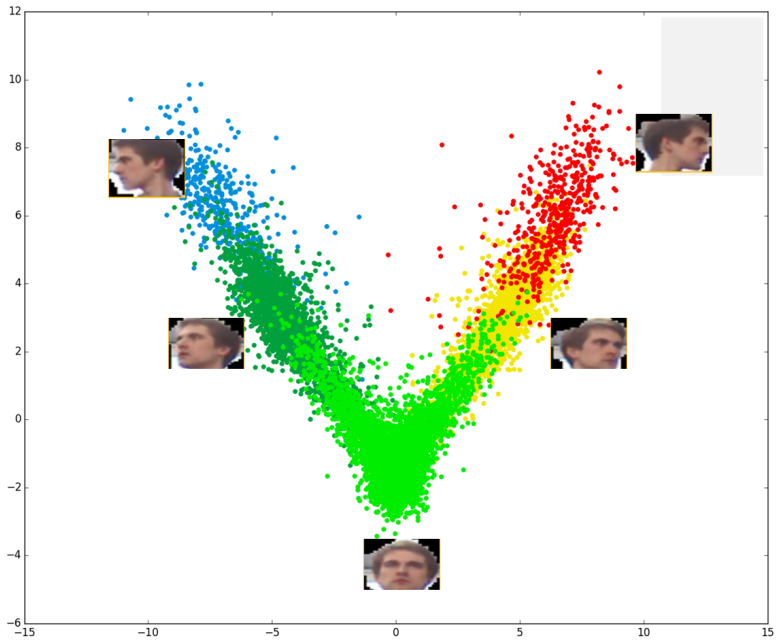



FIGURE 4.14: Linear Discriminant Analysis (LDA) projected scatter plot of the regression features on the BIWI dataset [1]. This shows the headpose regression features thinly form around the frontal pose manifold. This dataset is relatively easy as it is very high resolution and contains only frontal/near-frontal head poses.

however we do not encode the inferred gravity (vertical) direction, because in our case the heads are always upright and this parameter would yield no more information. We do however encode the surface normal azimuthal angle and the surface normal elevation angle along with the depth data to form three channels as can be seen in Fig. 4.9.

Surface normals have proved to be a very useful feature for object recognition [113]. We compute the surface normals via:

$$\overrightarrow{N_{X_i,Y_j}} = \frac{\overrightarrow{\partial_X Z_{ij}} \times \overrightarrow{\partial_Y Z_{ij}}}{\left\| \overrightarrow{\partial_X Z_{ij}} \times \overrightarrow{\partial_Y Z_{ij}} \right\|} \tag{4.7}$$

where $\overrightarrow{N_{X_i,Y_j}}$ is the normalized normal vector at $X_i, Y_j, Z_{ij}$ which in turn are the real world coordinate at depth image point $U_i, V_j$and $\overrightarrow{\partial_X Z_{ij}}$ is the X derivative and $\overrightarrow{\partial_Y Z_{ij}}$the Y derivative at point $X_i, Y_j$. To compute the derivatives we use implicit filtering techniques as described in [108, 123]. Implicit filtering techniques are much more accurate than the standard morphological derivative as can be seen in Fig. 4.15. Implicit filtering also involves larger neighbourhoods for computing more accurate gradients. Considering all these benefits we chose to use the one

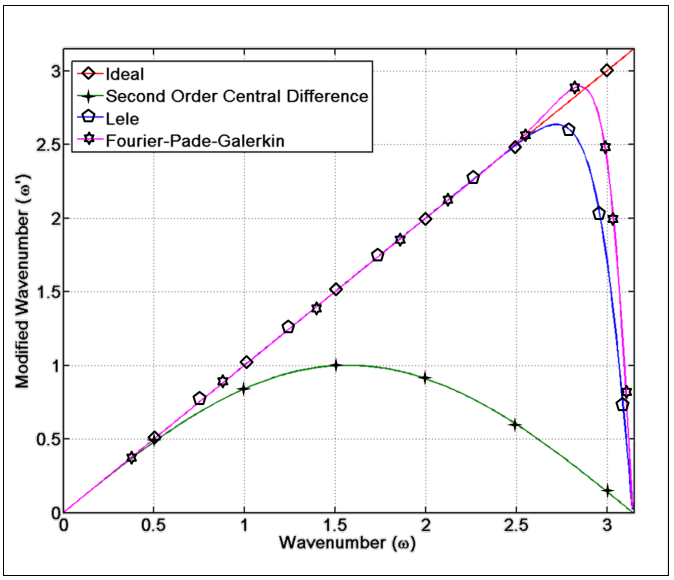



FIGURE 4.15: The benefit of implicit differentiation is shown in this graph. It can be seen that the frequency response of the implicit Lele's and Fourier-Pade-Gelerkin schemes [108] better approximate the original derivative compared to the explicit second order central difference scheme.

parameter family of implicit differentiation with the frequency domain transfer function defined as the following spatial domain equation:

$$\beta f'_{i-2} + \alpha f'_{i-1} + f'_i + \alpha f'_{i+1} + \beta f'_{i+2} = c\frac{f_{i+3} - f_{i-3}}{6h} + b\frac{f_{i+2} - f_{i-2}}{4h} + a\frac{f_{i+1} - f_{i-1}}{2h} \quad (4.8)$$

where $f'_i$and $f_i$ are the $x$ derivative and the function value respectively at $x = i$ . and its corresponding frequency domain counterpart is defined as follows

$$H(\omega) = j\frac{a \times \sin\omega + (b/2) \times \sin 2\omega + (c/3) \times \sin 3\omega}{1 + 2\alpha\cos\omega + 2\beta\cos 2\omega} \quad (4.9)$$

where $\alpha, \beta, a, b, c$ are user chosen parameters.

The $y$ derivative can be computed similarly. There are many standard parameter choices for a,b,c, $\alpha$, and $\beta$. Here we use the Lele coefficient values [108].

### 4.2.3 Fine-tuning for regression

The classification network is turned into a regression network by replacing the last Softmax layer with an Euclidian loss layer that measures the L2 distance of the prediction from the target. To activate the fine-tuning on regression on the head-pose data the following must be considered. The regression problem is ill-posed for the linear Euclidean manifold where we compute the regression L2 loss. This is because the normalized regression label goes from 0 to 1 where 0 is the back of the head to 0.5 that is for front facing to 1 (360 degree) that is again back of the head. Now the distance between the angle 0.1 and 0.9 should be 0.2 on the circular manifold. In the stated example the heads look very similar, however the loss function penalizes the network by having an error of 0.98, hence the gradients for weight update are large and these force large changes. Ideally the loss function would be defined as:

$$L = \begin{cases} \frac{1}{2}(t-o)^2 & if\ t-o < 0.5 \\ \frac{1}{2}((1-(t-o))^2 & if\ t-o > 0.5 \end{cases} \tag{4.10}$$

where t is the target angle and o is the output of the network. However this function is not everywhere differentiable (with a discontinuity at $t - o = 0.5$). In order to perform gradient descent the loss must be differentiable w.r.t the weights. To overcome this issue, instead of using the angles for regression, we use the X,Y coordinate of the unit vector pointing in that angle, the problem can be posed on the linear Euclidean manifold again. So instead of a single number we have a pair. For both Yaw and Pitch this same technique can be easily extended to use the X,Y and Z coordinates of the head pose vector in 3-D. The network fine-tuned for regression should have features that are thinly spread along the manifold. We see this expected result in Fig. 4.14 where we plot the features projected to two dimensions using Linear Discriminant Analysis (LDA) on the Biwi dataset[1] . We also plot the same using our dataset in Fig. 4.13.

### 4.2.4 Fusion of RGB and Depth modalities

Whenever available, both RGB and depth give complementary information that can be combined to achieve an overall information gain. Apart from that depth information can further be exploited to compute the scene interaction/ attention

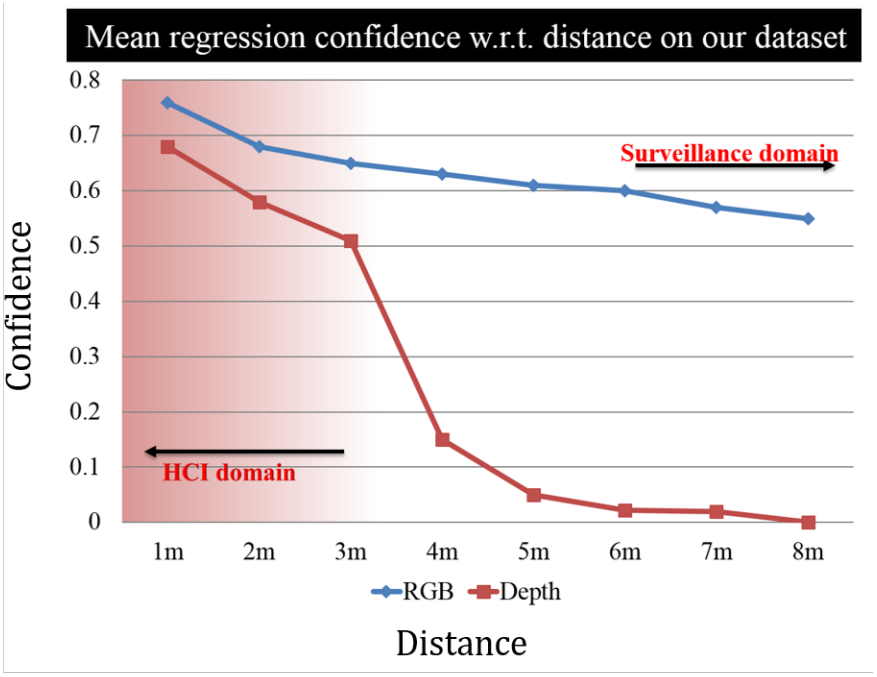



FIGURE 4.16: The quality of depth data degrades rapidly while RGB stays much more reliable with distance

metric in 3D that maps the head pose based attention to the 3D environment. Hence whenever possible, we average the class posterior scores from both the RGB and depth classifiers. However we note that depth information quality is highly dependent on distance from the sensor. Also from our experiments we have found out that the back of the head depth images are extremely noisy.

Figure 4.16 shows the reliability of the RGB vs Depth information as a function of distance from the sensor (in this work we used both the Kinect and Kinect v2 sensors). We compute the confidence of the RGB and Depth information from the relative error with respect to ground truth. From our experiments we have seen that unless the distance of the detected head is taken into account, depth information is not very reliable after 3.5 metres as far as headpose is concerned. Hence if depth data is available and the detected head is less than 3.5m distant, we average the output of the RGB and Depth models. Otherwise we only use the RGB information (e.g. in the surveillance domain). As can be seen from Fig. 4.17, depth information also degrades rapidly for non frontal poses but is very useful close HCI domain data.

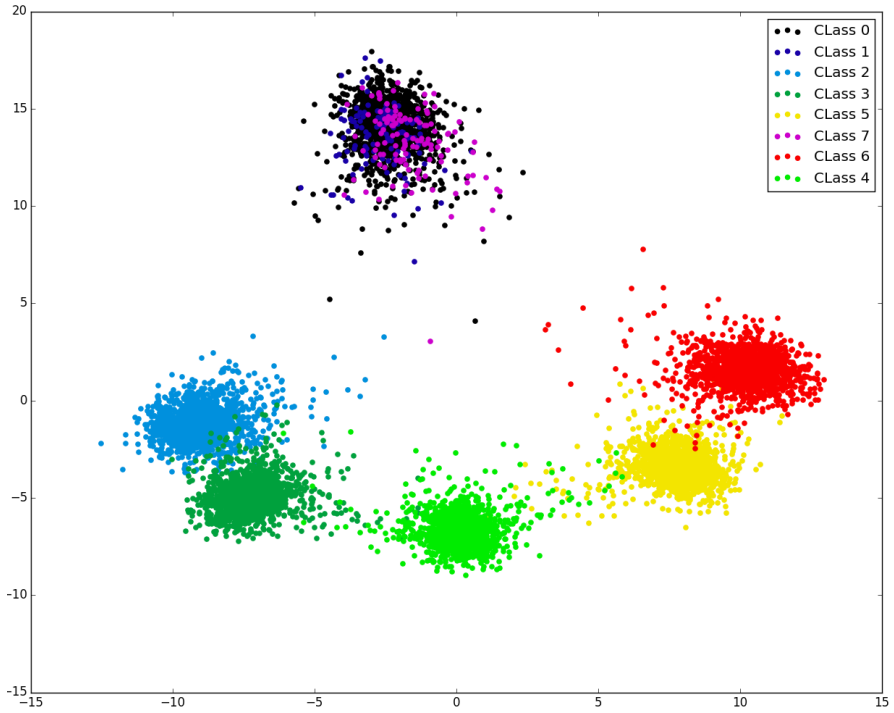



FIGURE 4.17: LDA scatter of depth information shows that it is not very reliable for back of the head poses. This is in line with our expectation as hair does not reflect the depth sensor Infrared illuminant very well and this often results in very noisy and sparse data. However the depth information is quite good for the frontal poses

### 4.2.5 Regression confidence estimate

We determine the regression confidence by combining the regression angle output on the yaw angle with the classifier posterior on the angles. For this we train a Softmax classifier with a granularity of 1 degree (360 classes) on top of the final regression network while keeping the rest of the network weights constant. This enables the computation of the variance of the posterior to estimate the confidence of the regression. Fig. 4.18 shows the output of the classifier posterior along with regression.

### 4.2.6 Experimental setup

We train one network for RGB and Depth each. This is done to unify the problem of both HCI and surveillance domains. Typically, one might adapt the networks for each domain, however from our initial experiments we found that including both high and low resolution imagery in the training set improved classification

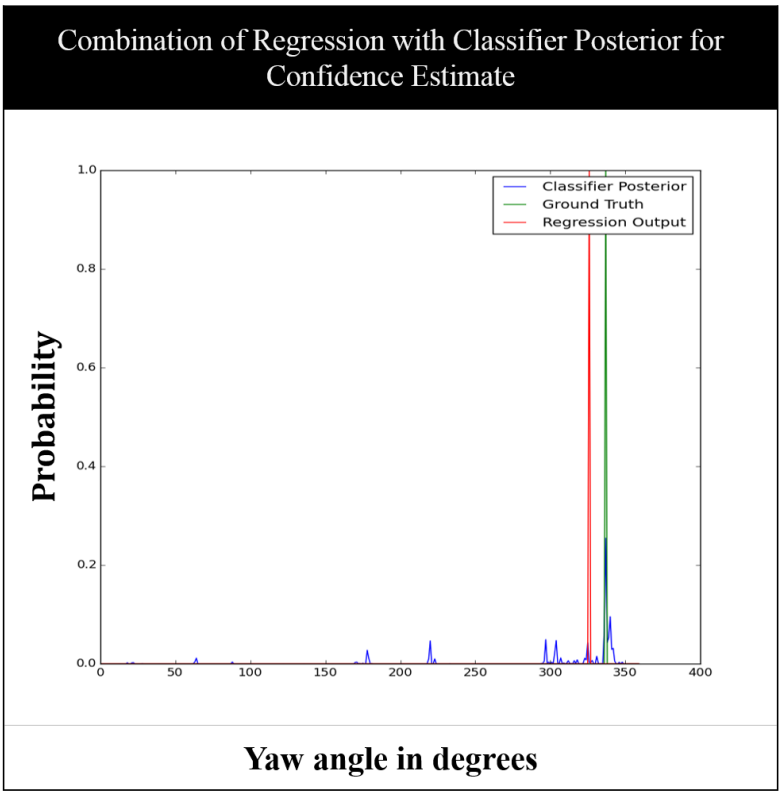



FIGURE 4.18: To estimate the confidence of the regression to model into the Attention metric we use the same network to get a posterior distribution on the 360 degrees. We show the regression output, the ground truth and the probability distributions

performance on the low resolution inference while the high resolution inference results were more or less the same. The convergence rate during training was faster as well. We think this is because the high resolution images help the network estimate the underlying model better and that translates into better parameter estimation for low resolution and/or noisy images. For training and validation we split our dataset in a ratio of 70:30 randomly across several trial runs and averaged the mean squared error. For training we used a dropout rate of 20% on before every fully connected layer. We jittered the input images by mirroring them (with corresponding change in groundtruth) scaling the bounding box and cropping them with scales 0.75, 0.9, 1.5, 1.8, 2.0, and 2.5. For all scales greater than 1, we also translated the images randomly by 20% in both directions. This was done to improve scale invariance along with mitigating the effects of poorly-aligned or partially-occluded head detections. We used a modified version of the deep learning framework Caffe [124] to train our network. We translated the centroid of each head to $(0, 0, 0)$ in 3D Euclidean space and uniformly re-sampled the point cloud to an organized $256 \times 256$ set. For re-sampling we used bi-cubic

interpolation for the RGB values and nearest neighbour interpolation for the XYZ values. To obtain the mean inherent variance due to eye balls (the true focus of attention is somewhat independent of head pose), we set up an experiment with where we tracked the difference between the absolute head-pose (using the IMU) and the focus of attention of the eye using the Gazepoint eye tracker, which has a resolution of 0.5° degrees at upto 30 cm distance. We computed the mean variance for 11 people. This provides us with an interesting insight to the problem. In conclusion, head-pose error less the mean error of 12.35° does not make any sense for the application of *true visual attention* estimation without tracking the additional dimensional freedom provided by the eyeballs. In order to gain a good understanding of human attention model without eyeball tracking, further studies into human gaze pattern with respect to scene saliency and semantic contextual information would be needed.

We selectively fused the RGB and Depth modalities based on availability and quality of the depth data as shown in Fig. 4.16. We only fuse the depth classification if the detected head is less than 3.5 metres in distance, otherwise the reliability of the depth data falls off rapidly as can be seen directly from the lower curve in Fig. 4.16.

For all the experiments we trained both our RGB and Depth CNN by randomly selecting 70% of the samples from our own dataset. This was due to the fact that deep CNNs require significantly more training data than traditional approaches. However, by fixing the training data, we get unbiased results for all the other datasets in validation. Unlike the other methods that partition the same data for training and validation, our approach shows the generalisation of our technique to unseen datasets.

Here we present the comprehensive validation of our technique on both HCI and surveillance domains.

### 4.2.7 Validation on BIWI Kinect Headpose Dataset [1]

The data in this dataset has been captured very close to the sensor and does not contain non-frontal poses. Here the output of our RGB and Depth models are averaged to get the result. This data lets us compare our general technique to a finer grained HCI technique as presented in [1]. The comparative results

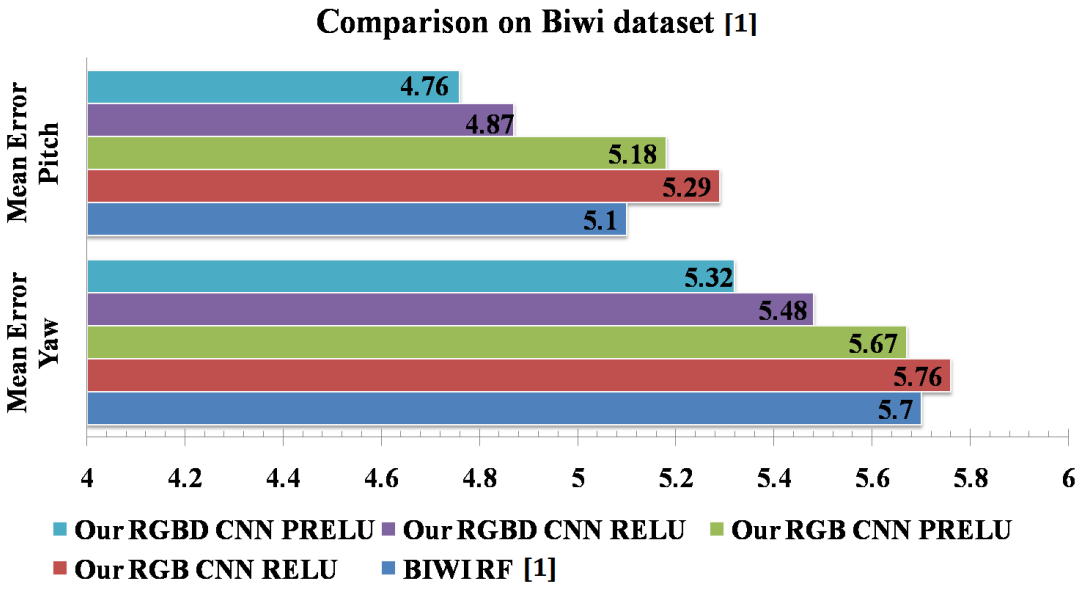



FIGURE 4.19: Comparison of our method on the BIWI dataset with respect to the random forest (RF) algorithm [1]. Our RGB+Depth CNNs (both RELU and PRELU) outperform the HCI technique without explicitly tracking facial landmarks. Here in this range we see the tangible benefit of having depth information along with RGB data.

are shown in Fig. 4.19. We use mean angular error as the metric which is the same used in comparable literature [1]. In both pitch and yaw we outperform the best method [1], which has the advantage of explicit landmark detection, by 7%. It should be noted that while we do not detect landmarks explicitly, from Fig. 4.11, it is clear that the CNN has now learned landmark detection automatically. However as can be seen from Fig. 4.11 the network detects landmarks whenever necessary implicitly along with other non obvious features. We also see that depth information actually improves the results in this range when combined with RGB.

### 4.2.8 Validation on our dataset

One weakness of the BIWI dataset is that it does not contain non frontal or distant head-pose data. To overcome this, and to show the power of our technique we report the results obtained on our dataset which is far more challenging. Our two baseline methods are the "Here's Looking at You Kid" (HLYK) [28] which uses only the RGB data and the Random forest (RF henceforth) based approach [1] which uses only the depth and normal data. We outperform both the techniques

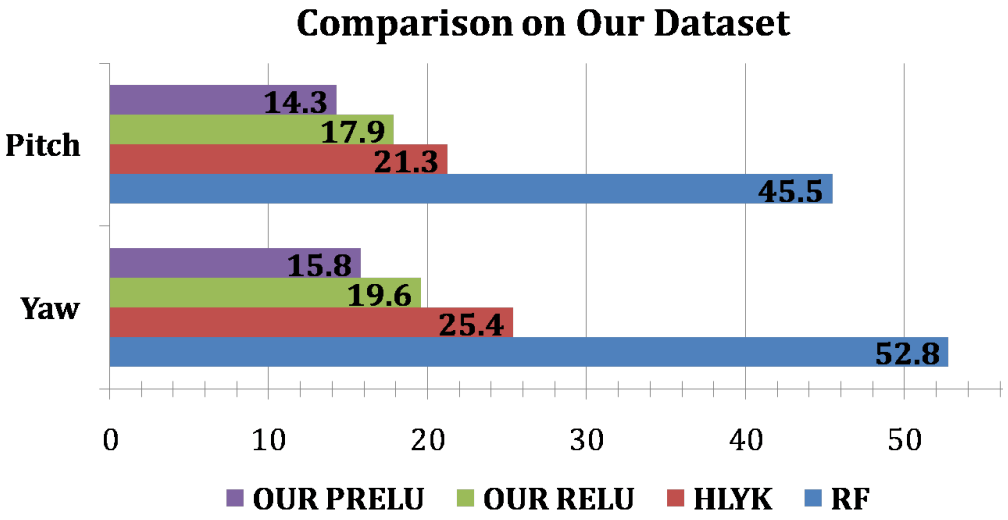



FIGURE 4.20: The mean squared errors (MSE) of the RF [1] and HLYK [28] techniques compared to ours on our dataset

by a significant margin as shown in Fig. 4.20. We reduce the relative error by 40% to that of our closest competing technique [28].

### 4.2.9 Validation on low-resolution surveillance dataset

For the low resolution surveillance domain dataset, we report our results on the Oxford and the Caviar datasets. In these datasets we classify the head pose into 8 equally spaced (45°) angular bins as shown in Fig. 4.12. For comparison with [11] and Benfold [116] we use the Oxford dataset in which both have reported results. One consideration has to be made while comparing because [11] reported the mean square error (MSE) which they derived from a weighted combination of their 8 class classifier output multiplied with the bin angles as $\sum_{i=1}^{8} p_i \overrightarrow{\eta_{\theta_i}}$ where $p_i$ is the classifier output value for the class $i$ and $\overrightarrow{\eta_{\theta_i}}$ is the unit vector in that angular direction. Fig. 4.21 shows the comparison between our method with the previous state-of-the-art results. In terms of MSE we have achieved the best published results. The margin alone does not give the true picture of performance. We therefore present the confusion matrices on the Oxford and Caviar datasets, as shown in Fig. 4.22.

On the Oxford dataset, for comparison, we also show the output confusion matrix of the Benfold algorithm [10] along with our confusion matrix as shown in Fig. 4.22.

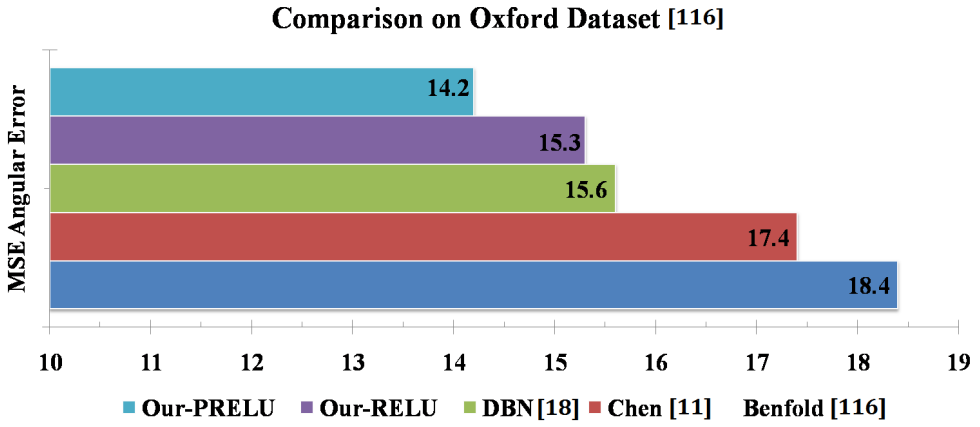



FIGURE 4.21: Mean squared error on the Oxford dataset. Here we compare our regression output with the Benfold [116] and the Chen [11] techniques

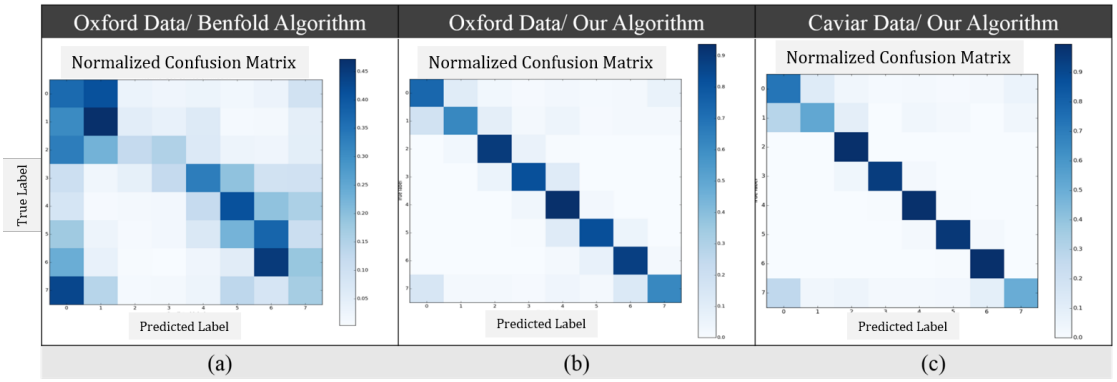



FIGURE 4.22: Confusion matrix comparing the methods the results. (a) Oxford data benfold algorithm [116], (b) Oxford data, Our RGB CNN, (c) Caviar data our RGB CNN. On both the datasets we have by far the state-of-the-art results

TABLE 4.1: Results on the Multi-PIE dataset

| Method | Mean Angular Error in ° |
|---|---|
| LDQP[32] | 7.3 |
| circ23D[30] | 5.8 |
| Our RELU | 4.56 |
| Our PRELU | 4.2 |

### 4.2.10 Validation on Multi-PIE dataset [2]

The Multi-PIE dataset consists of 337 subjects, under 15 view angles and 19 illumination conditions. This is a close range high resolution RGB dataset. We compare our method against two state-of-the-art techniques on this dataset 1. LDQP [32] and 2. circle23Sphere [30]. As shown in Table 4.1 we outperform both

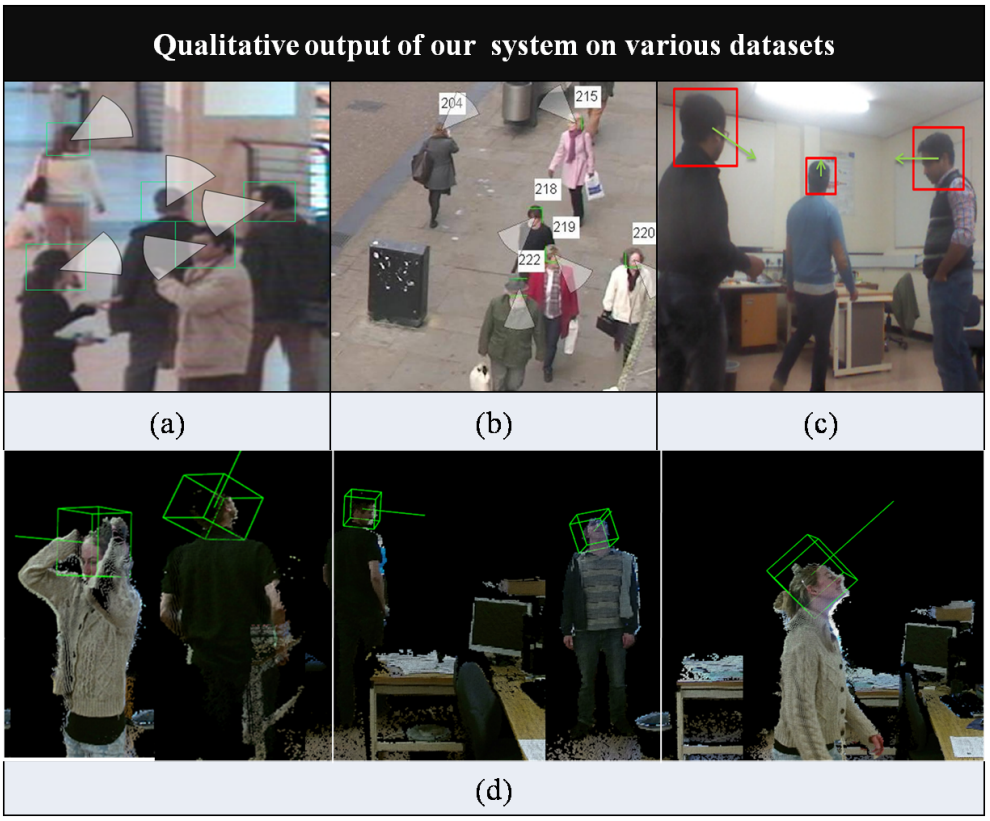



FIGURE 4.23: Qualitative output of our headpose estimation system on various datasets. (a) Caviar dataset, (b) Oxford Dataset, (c) Our RGB dataset, (d) Our low resolution RGB-D dataset.

competing techniques [32] and [30] in terms of Mean Angular Error(MAE) by a significant margin without any training on this dataset.

### 4.2.11 Comparison between PRELU and RELU

In all our experiments PRELU activation outperformed RELU consistently. As can be seen from Figure 4.21, in case of low resolution surveillance domain dataset [5] , we gain $> 1°$ improvement in angular error by using PRELU. On our challenging RGB-D dataset the effect is even more pronounced with an improvement of around $2.9°$ as seen in Figure 4.20. Results on the CMU Multi-PIE dataset as reported in Table I suggests that PRELU provides an additional reduction in MAE of $0.36°$. Finally in Figure 4.19, on the Biwi dataset where error rates are already pretty low, we get less improvement ( $0.2°$), but consistent improvement nonetheless by using PRELU over RELU.

## 4.3 End-To-End Head Detection and Headpose Estimation

In this section we propose a Fully Convolutional Neural Network (FCNN) based approach for end-to-end head detection and pose estimation called Detection Fully Convolutional Neural Network (DFCNN). Traditionally, CNN based detectors like Deformable Part Models [125], R-CNN [75], Faster RCNN [78], YOLO [126] learn to be invriant to pose and other geometric transformations of an object. Hence the features from the detector, is not informative for pose. On the other hand, classification networks when applied in a fully convolutional fashion, like the FCN-8s [42] retains for localisation and classification properties. However, these semantic segmentation networks can not delineate between instances of the same object. To overcome these drawbacks, we propose a new fully convolutional architecture that excels at both the tasks. As an added benefit, the network shares the whole computation on an image thus making it a lot more efficient compared to the detections networks like R-CNN [75], and Faster RCNN [78].

We also introduce a new loss function called the Adaptive Localizing Infogain Cross Entropy (ALICE) loss to convert the detection and pose classification to a dense classification problem with weak labels (normally only the head bounding boxes are labelled instead of of the segmented head and its pose class). We report near state-of-the-art result for detection on the Hollywood dataset[3] and state-of-the-art result on the Oxford Town Centre Dataset [4] dataset. We also achieve state-of-the-art result in head pose classification.

### 4.3.1 DFCNN Architecture

We adapt the VGG-16 architecture [127] which has been succesfully used for both detection[75] and dense semantic classification[42]. We apply the a trous trick to the last 3 convolution layers, that uses dialated convolutions to increase the resolution of the features that are downsampled in max-pooling layers. This has been shown to improve sptial localisation performance [96]. We first train a 8 class model that spans the 360° and a background class (for non head patches sampled arbitrarily from the Hollywood heads dataset). We also add a a 4 real number regression layer to predict the corners of the head bounding box in the patch in

normalised coordinates. We then convert the last 3 fully connected layers into convolution layers [42]. This helps us apply the neural network to arbitrary sized image in a dense fashion. This produces a 13 channel output map. $NC + 1 + 4$ channels, where $NC$ is the number of headpose classes, in our case 8, 1 for the background class, and the 4 channels for the distance to the closest corresponding corner to the bounding box.

### 4.3.2 DFCNN forward-backward propagation, and inference

The DFCNN architecture can be described in three parts.

- *Encoder frontend.* This fully convolutional network with the converted inner product layers, converts an input image of dimensions $W \times H \times 3$ to $W/32 \times H/32 \times 512$, where W and H are the width and height of the input image, 3 is the number of channels for the the input RGB image, and 512 is the number of feature channels at the end. This part maps the image into a feature cube that is a compact but discriminative representation of the image. There are 5 non overlapping max-pooling layers with size 2. Hence the input gets downsampled by a factor of 32.
- *Upsampling.* This part maps the $W/32 \times H/32 \times 512$ feature volume back to the image dimensions with a deconvolution layer. The number of deconvolution filters is equal to C+1+4. This is for the C classes, which for our case is 8, 1 background class, and 4 channels for the bounding box. The output of each of the channel classes is a per pixel probability map of the pixel belonging to that class. The 4 channels for the bounding box is the pixel map for the distance to the nearest bounding box coreners for all the four corners of a rectangle.
- *Slice and Inference.* This is the final part of the network that slices the previous $W \times H \times (C + 1 + 4)$ volume into two semantically different volumes of $W \times H \times (C + 1)$ for the class map and $W \times H \times 4$ for the bounding box map. In the training phase the classmap is trained with the ALICE loss and the bounding box map is trained with the L1 loss. The gradients of these loss are equally added and backpropagated through the network.
  During inference a standard pretrained region proposal network is appended

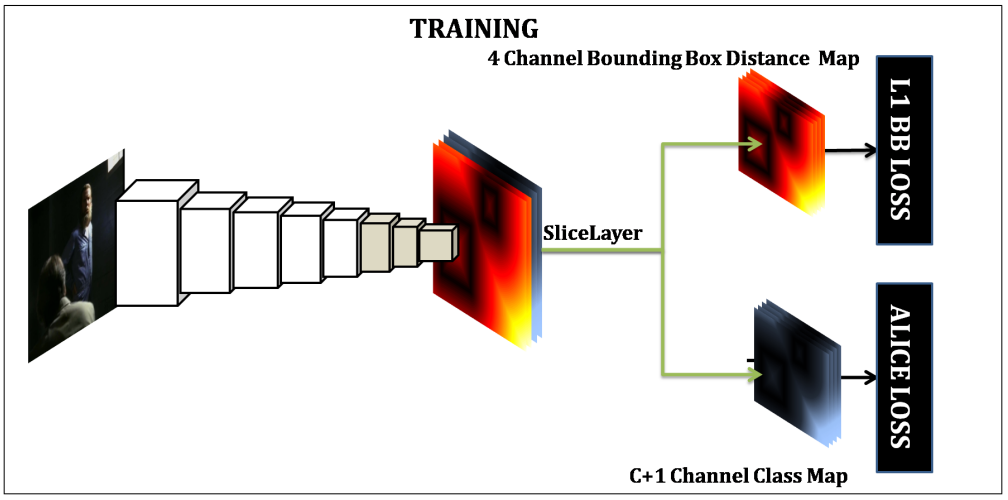



FIGURE 4.24: Architecture for training the DFCNN. The network is pretrained on pose classifications and then converted to a FCNN. Then a final convolution layer is added to add the bounding box output map pose classification map. The two tasks are trained in parallel with a joint multi-task loss.

after the encoder stage, in parallel to the Upsampling and Slice and Inference stages. This region proposal network (RPN) produces class agnostic bounding boxes. For each of the bounding boxes we average the class probability represented within the bounding box. The maximum of the averages of the classes is selected as the class. For regions with no heads the average is the backgroud class. For the other remaining bounding boxes, we use the 4 channels of the bounding box map to average the distance to each of the closest 4 vertices inside the predicted boxs from the RPN network. We then correct the RPN box proposal with the average distance from the box maps. Finally we do a non maximal suppression of the overlapping boxes with Intersection-over-Union (IoU) ratio greater than 0.4. This produces a list of boxes for human heads along with their headpose classes

### 4.3.3 Training

During training we slice the classification and regression layers feed them into two loss functions. For the classification we use the ALICE loss as described in section 4.3.5. To generate the training data for the ALICE loss we compute the distance transform inside each bounding box from the centre of the box to weight the strength of the loss strength. This means that when the ALICE loss is evaluated densely within the bounding box region, the class of the headpose within

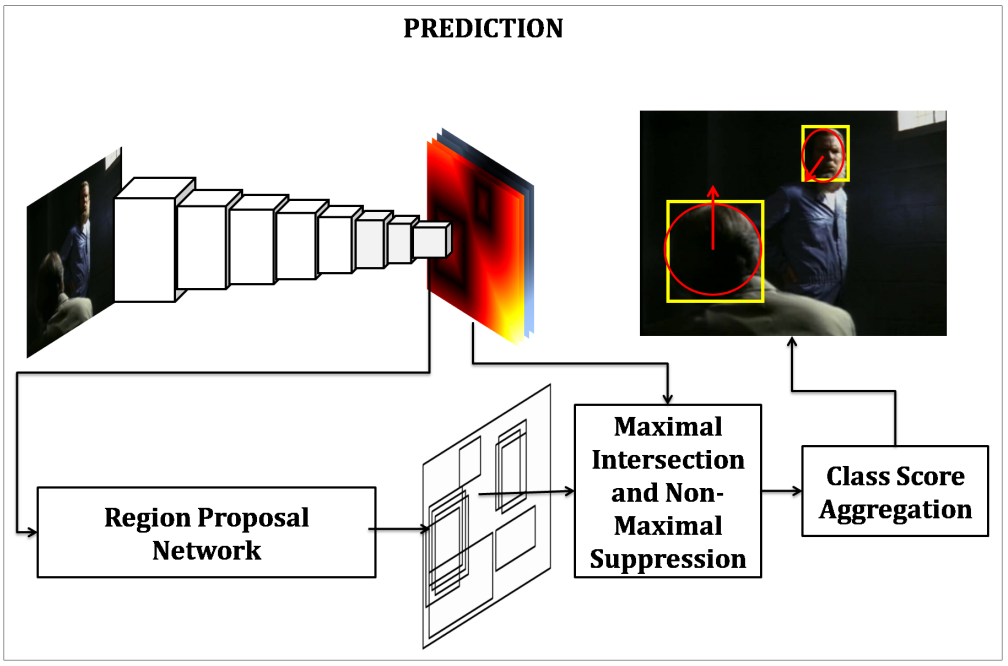



FIGURE 4.25: Architecture for prediction using the trained DFCNN. The output maps and the region proposal network outputs are combined using maximal intersection and non maximal suppression to get final bounding boxes. For each bounding box, the dense pose class is average pooled within the region to get the pose.

that bounding box is evaluated with all the pixels within that box. We observe that heads are circular in appearance, hence, the loss strength is maximum at the centre and isotropically drops off as it goes towards the edge. We control the loss strength further with the help of the $\gamma$ and $\tau$ parameters as decribed in section 4.3.5.

For the bounding box regression we use the L1 loss defined as

$$E = |x - x_p| \tag{4.11}$$

Where $x$ is the ground truth distance of the corresponding corner of the bounding box from the pixel being evaluated and $x_p$ is the predicted output. We normalize the coordinates to the image size so that the values lie within $0 - 1$ range. Figure 4.24 shows the training procedure and architecture.

### 4.3.4 Prediction

Once the neural network is trained, we use it along with a pretrained region proposal network from [78]. This predicts possible object locations.

For each predicted bounding box we average the bounding box distance map values to get the average predicted bounding box within that location which we call maximal intersection. Finally we apply non maximal suppression to obtain the final head bounding boxes. For predicting the headpose within the bounding box we use a region based average pooling to get average headpose enclosed by that bounding box. Figure 4.25 shows the prediction architecture.

We bring to attention two factors here. One, by eliminating the need to run the bounding box features through the last fully connected layers repeatedly like in [78] we gain a massive computational advantage that is proportional to the number of bounding bxes predicted by the region proposal network. Second, by averaging the headpose prediction within the bounding box, the architecture should be more robust to context and occlusions.

### 4.3.5 Adaptive Localizing Infogain Cross-Entropy Loss (ALICE)

We define the ALICE loss as the pixel weighted cross-entropy loss with class information gain as follows

$$L(l^{(n)}, p^{(n)}) = \frac{-1}{N} \sum_{n} \left[ \sum_{k=0}^{H-1} \left[ \sum_{j=0}^{W-1} \left[ \sum_{i}^{m} H_{lc} S_{ijk} l_{ijk}^{(n)} \log p_{ijk}^{(n)} \right] \right] \right] \tag{4.12}$$

Where $l^{(n)}, p^{(n)}$ are the true label and predicted probability over labels respectively. The loss is defined over the entire image with indices j, and k summing over the image. To simplify the notation we henceforth drop the j, and k indices and define the loss at every pixel as

$$L(l^{(n)}, p^{(n)}) = \frac{-1}{N} \sum_{n} \sum_{i}^{m} H_{lc} S_i l_i^{(n)} \log p_i^{(n)} \tag{4.13}$$

where n is the number of images in the training mini-batch. $l_i^{(n)}$ is the true label, $\log p_i^{(n)}$is the log probability of the output of the network. The $H_{lc}$ is the information gain term that is drawn from a matrix of size $m \times m$ that defines the relative strength of the loss given class label $l$ and class prediction $c$ where $m$ is the number of classes. The term $S_i$ is the multiplier for the strengh of the loss given the distance of the pixel from the closest ground truth label. The motivation for the different terms are described next.

**Adaptive.** We initialize the H matrix according to the class imbalance present in the dataset, but we do not fix the H matrix during training, instead we let the solver optimize the H matrix as follows

$$H_{lc} = H_{lc} - \eta \frac{\partial L}{\partial H_{lc}} \tag{4.14}$$

The intuition behind is that depending on the problem at hand, class imbalance may not be the only determining factor for performance. Also this helps the back headposes to be well separated in feature space. Hence, by letting the solver optimize the H matrix as it goes along, it will be able to directly optimize the confusion matrix to be diagonal, instead of optimizing for overall accuracy. The only constraint we impose on the H matrix is that after each mini-batch update we normalize it by dividing it by its determinant

$$H = \frac{H}{|H|} \tag{4.15}$$

This ensures that the optimisation of the $H$ does not produce unbouded resutls. We also fix the learning rate $\eta$ of the $H$ matrix to 0.0000001.

**Localizing.**Since our ground truth is bounding box based we would normally evaluate the ground truth inside the bounding box locations ignoring the rest. However this introduces error between head and background/occlusion. Hence we generate a pseudo label inside the bounding box by using distance transform [128] from the centre of the box to the edge. The strength of the loss back propagated is multiplied by the strength of this pseudo label. That is at the centre of the box,

it takes propagates the loss with a multiplier of 1 and it falls off as we go towards the edge of the box. This helps us give the classifier the context of the whole head using a pseudo label. We control the strength of the pseudo label by using a $\gamma$ parameter defined as follows.

$$S_i = AI^{\gamma} \tag{4.16}$$

where $I$ is the distance transform based on the true class labels.

$$I_{out} = \begin{cases} I_g & I_g \geq \tau \\ 0 & I_g < \tau \end{cases} \tag{4.17}$$

where A is the gain parameter and is set to 1 for our cases. We vary the $\gamma$ and $\tau$ and find out optimal values for them through cross-validation during our experiments.

### 4.3.6 Training and Validation

Although, there are multiple datasets for face detection, the only publicly available large scale dataset for head detection is the Hollywood Heads dataset [3]. However it does not have headpose annotations. On the other hand the Oxford dataset [4] has both head and headpose annotations, however it not large enough to train a full CNN end to end. Hence we use our CNN model trained in section 4.2 to assign a headpose class to every head in the Hollywood Heads dataset. We divide it into 45 degree bins to keep the annotations accurate.

For pseudo label generation we found the best $\gamma = 3.5$ using a step of 0.5 from $1 to 5$. We combined the datasets, for training and split them into $60 : 20 : 20$ ratio for training, validation and testing.

### 4.3.7 Results

In this section we validate our detection and heapose classification performance on two publicly available datasets the Hollywood heads [3] and the Oxford datasets [4]. For detection we also compare various other methods like DPM [125] RCNN

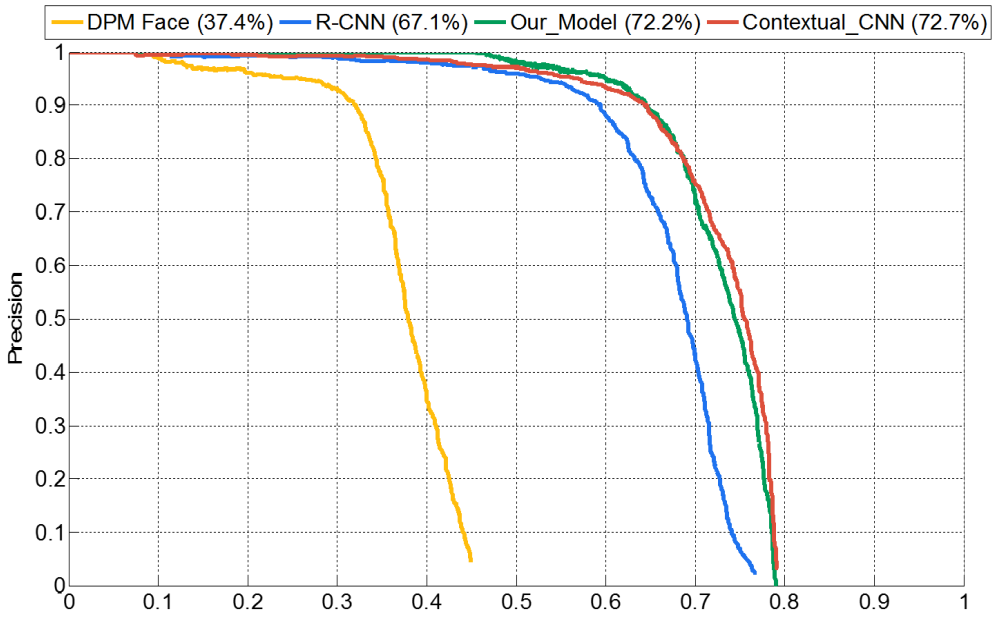



FIGURE 4.26: Precision-Recall curves for various detectors on the Hollywood Heads dataset [3]. We report the detectors DPM [125], RCNN [75], Contextual-CNN [3] and Our Model. We also report the average precision in the brackets. Here we see that our model achieves performance close to the state-of-the-art.

[75] and the Contextual-CNN [3]. For evaluating classification performance, we report the confusion matrix and the overall accuracy. To compare the effect of detectors we use the state-of-the-art Contextual-CNN [3] detector and our CNN model *Section*4.2 for pose classification and compare it to our end-to-end solution.

### 4.3.8 Validation on the Hollywood Heads [3] Dataset

**Detection performance.** We first compare the detector performance on the Hollywood heads dataset. We use the precision-recall curves as shown in Figure 4.26, to compare the various detectors. We also report the average precision metric.

**Classification Performance.** We compare the disjoint detector [3] , classifier as in Section 4.2 called Disjoint approach to Our end-to-end approach using the confusion matrices and the overall accuracy. Figure 4.27 shows the two confusion matrices. In terms of overall accuracy, the Disjoint approach achieves 86.2% overall accuracy whereas our end-to-end model achieves 91.1% overall accuracy.

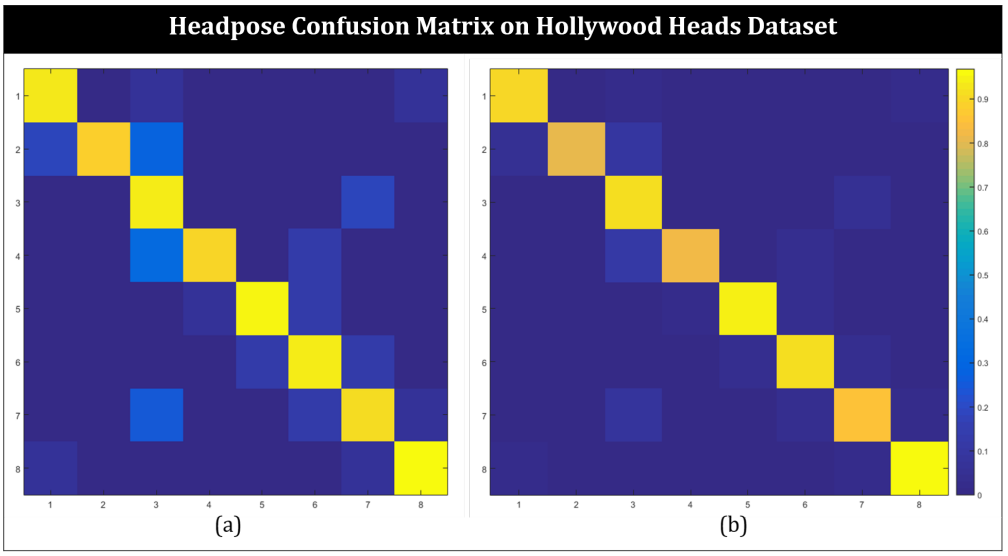



FIGURE 4.27: Confusion matrices on the Hollywood heads dataset [3]. (a) Depicts the result of using the Context model for detection and our CNN from Section 4.2 for classification. (b) Shows the end to end DFCNN approach. It is easily seen that end to end training improves performance.

### 4.3.9 Validation on the Oxford [4] Dataset

**Detection performance.** We first compare the detector performance on the Oxford dataset. We use the precision-recall curves as shown in Figure 4.28, to compare the various detectors. We also report the average precision metric. The heads in this dataset are a lot smaller in resolution and we achieve state-of-the-art performance. We believe this is due to the better field-of-view of the DFCNN model (which has a stride of 8)

**Classification Performance.**

We compare the disjoint detector [3] , classifier as in Section 4.2 called Disjoint approach to Our end-to-end approach using the confusion matrices and the overall accuracy. Figure 4.29 shows the two confusion matrices. In terms of overall accuracy, the Disjoint approach achieves 89.5% overall accuracy whereas our end-to-end model achieves 94.8% overall accuracy.

### 4.3.10 Qualitative Output

In Figure 4.30 we show the output of our end-to-end detection and headpose estimation framework on frames from the Hollywood Heads dataset.

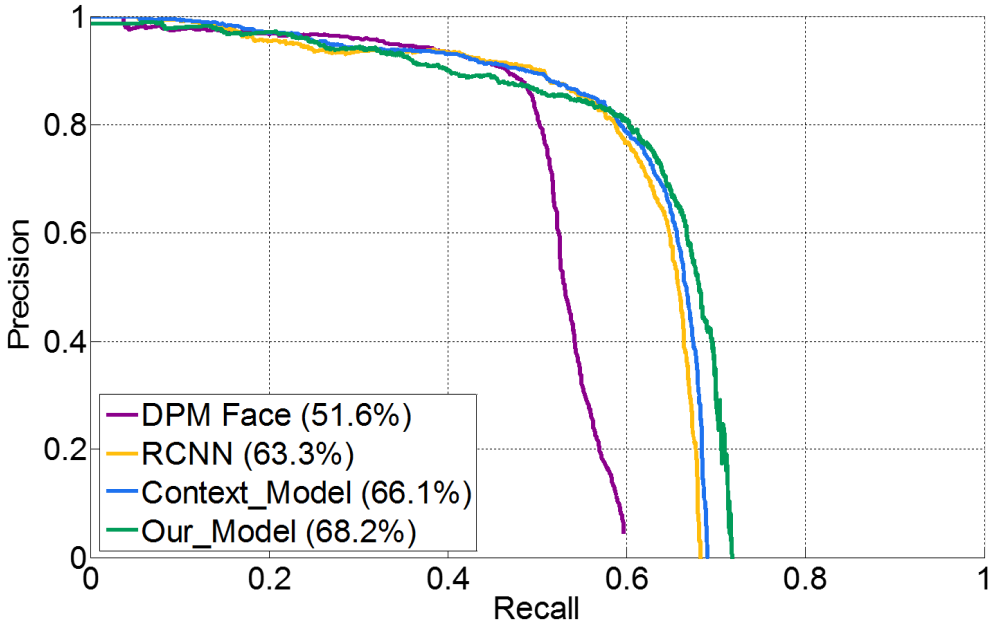



FIGURE 4.28: Precision-Recall curves for various detectors on the Oxford dataset [4]. We report the detectors DPM [125], RCNN [75], Contextual-CNN [3] and Our Model. We also report the average precision in the brackets. Our model achieves the state-of-the-art performance.

Figure 4.31 shows some of the errors of our end-to-end detection and headpose estimation framework on frames from the Hollywood Heads dataset.

## 4.4 Discussion

The importance of head pose as a separate independent information source has many applications. While coupling different priors like velocity and body direction may be good for bringing down MSE in a dataset, it actually attenuates the pure information content from the head pose. This has been apparent in an application where a standard Kalman filter [129] has been adapted to perform intentional tracking [8] which significantly improves upon the standard Kalman filter based trackers. Baxter et al. empirically showed that using head pose as a separate information source can be very useful, since many times detections are missed due to occlusions and it generates sub optimal velocity estimates in the Kalman filter. Moreover in case of these occlusions, neither the head-body coupling by Cheng et al. nor the velocity coupling of Benfold et al. will be of any use. Keeping these in mind and combined with the conjecture that people tend to look where they want to go before changing course, this intentional tracker has significantly

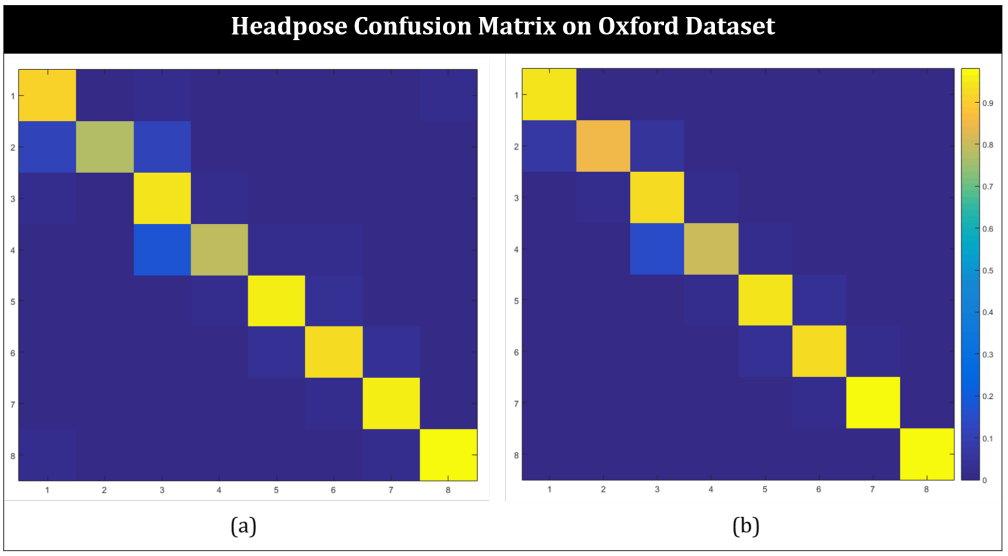



FIGURE 4.29: Confusion matrices on the Oxford dataset [10]. (a) Depicts the result of using the Context model for detection and our CNN from Section 4.2 for classification. (b) Shows the end to end DFCNN approach. It is easily seen that end to end training improves performance.

outperformed the standard Kalman filter based trackers. This uniquely shows the usefulness of a high performing robust independent only-visual head pose estimators like the one developed in this thesis. In the next chapter we show some applications of robust headpose estimation in some real world problems.

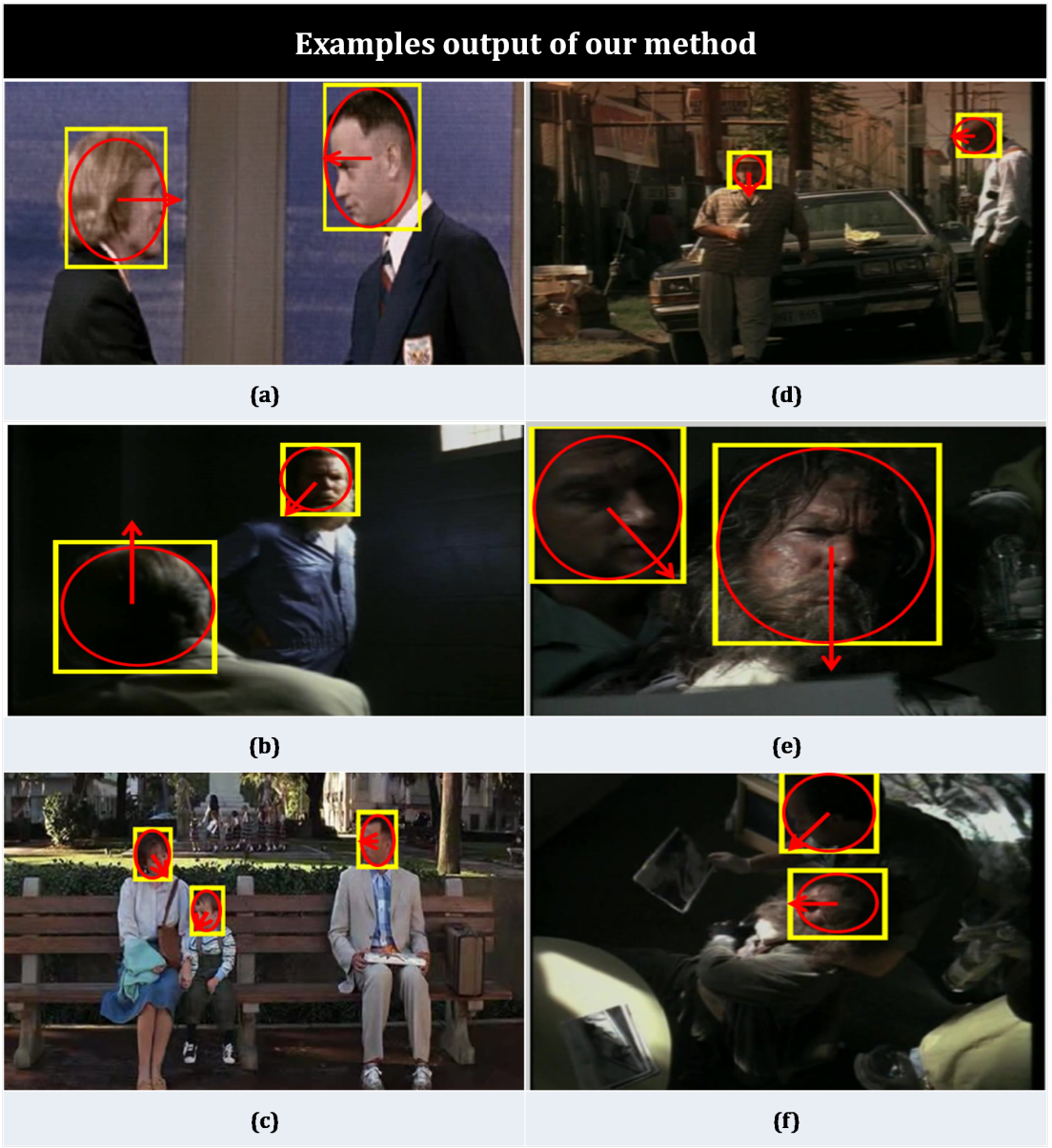



FIGURE 4.30: We show the output of our end-to-end detection and headpose estimation framework on frames from the Hollywood Heads dataset.

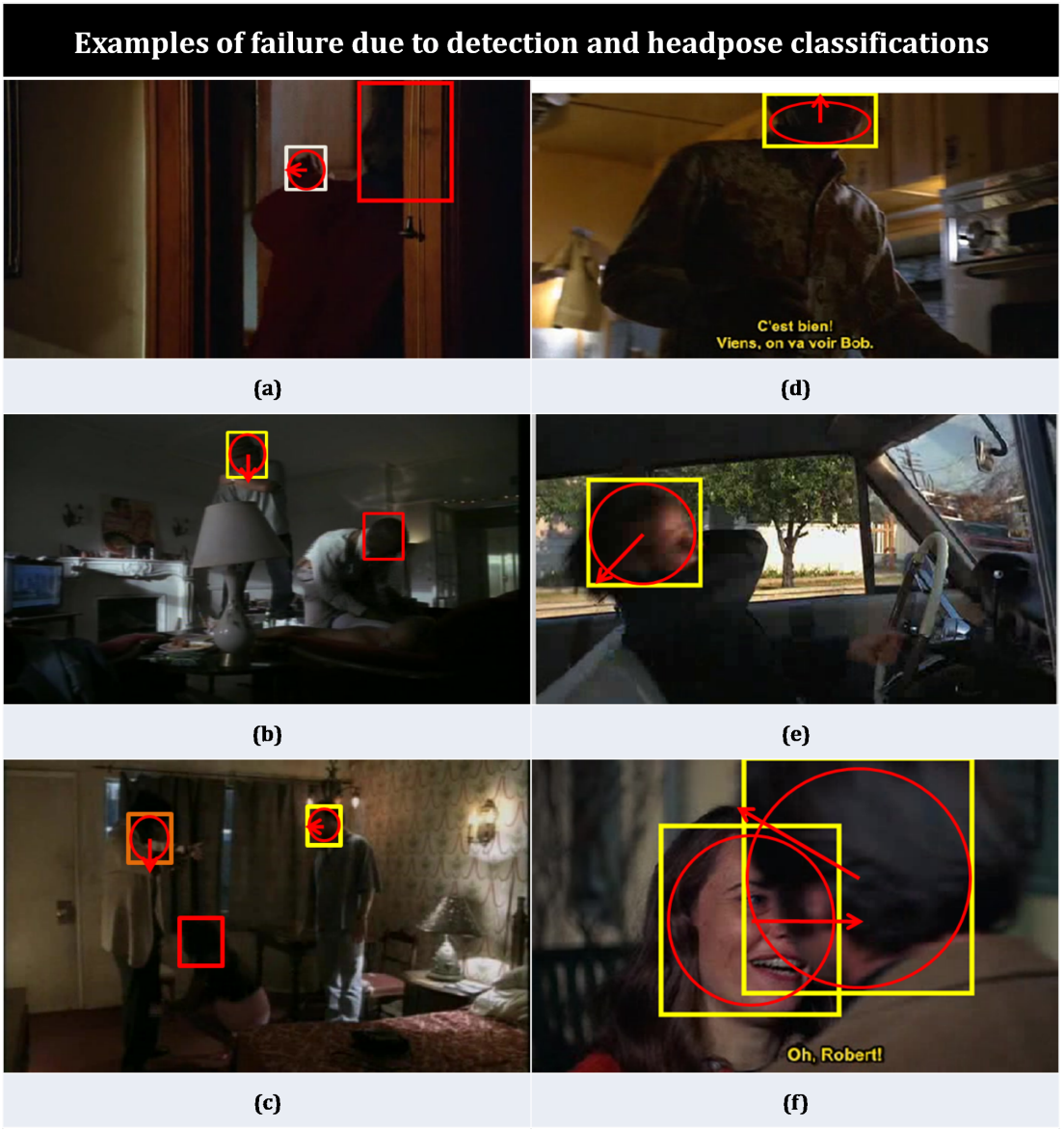



FIGURE 4.31: For completeness we show some of the errors of our end-to-end detection and headpose estimation framework on frames from the Hollywood Heads dataset. (a),(b) and (c) show misdetections that are drawn as red bounding boxes on the left column. As can be seen this is due to most very challenging lighting conditions. On the right column in (d),(e) and (f) we show bad headpose classifications which are mostly due to occlusion.

# Chapter 5

# Applications of Head Pose Estimation

*In this chapter we exploit the headpose as a signal for various applications. We first show that robust headpose estimation that is uncoupled from other prior informations like walking direction, body pose etc. can be used for various applications. First we show that, using this headpose signal we can improve the standard velocity model based Kalman Filter for tracking. This is called the intentional tracker. Furthermore we show that, this headpose signal can be used in social signal processing. We can detect human-human, human-scene interactions. We also develop two state of the art probabilistic metrics called the Attention and Interaction metrics for this purpose. We also show qualitatively, that a temporal windowed cross correlation computed between the headpose signals, give us an estimate of social mimicry that can be combined in future with other factors like spatial position and interaction metric to model human groups. The studies in this chapter have been published in the following: in IEEE Signal Processing Letters 2015 [8]; in IEEE Transactions on Multimedia (TMM) 2015 [16]; in VISAPP 2016 [17]*

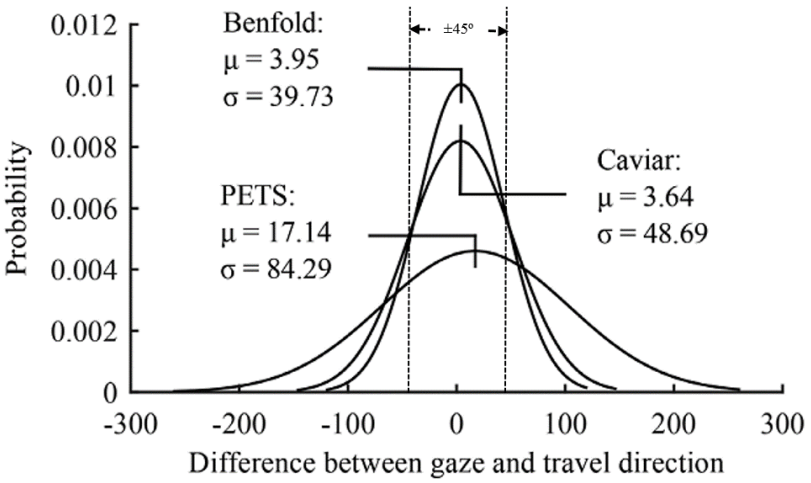



FIGURE 5.1: (a) Head pose deviation from walking direction as a Probability Density Function in various datasets [8] (b) The conceptual parametric human head space

## 5.1 An Intentional Prior for Kalman Filter Based Tracking

In visual surveillance the resolution of detected heads can be very small so head pose is often estimated in coarse discrete directional bins of the azimuthal angle [9]. See for example the eight classification bins used in this paper in Figure 4.12. Walking direction is then often used as a smoothing prior [4], which reduces mean squared error, but also attenuates the pure information content of the head pose signal. As shown in Fig 5.1, an analysis of gazing behaviour in several datasets demonstrates that most people look where they are going. However, the cases that are of more interest are when people deviate from this behaviour (i.e. look somewhere else), as this information could be useful for anomaly detection or improving tracking [8].

As an example, in scenes where there are dynamic obstacles and occlusions like parked vehicles and heave crowd occlusions, often times obstacle avoidance behaviour causes trajectory changes. Traditional techniques that learn a distribution of motion flows often fail when the occlusions and obstacles are dynamic[84–87]. To overcome these drawbacks, and with the observation that people tend to look where they are going before changing course, we formulate a novel headpose prior for the Kalman Filter provides better results in these scenarios.

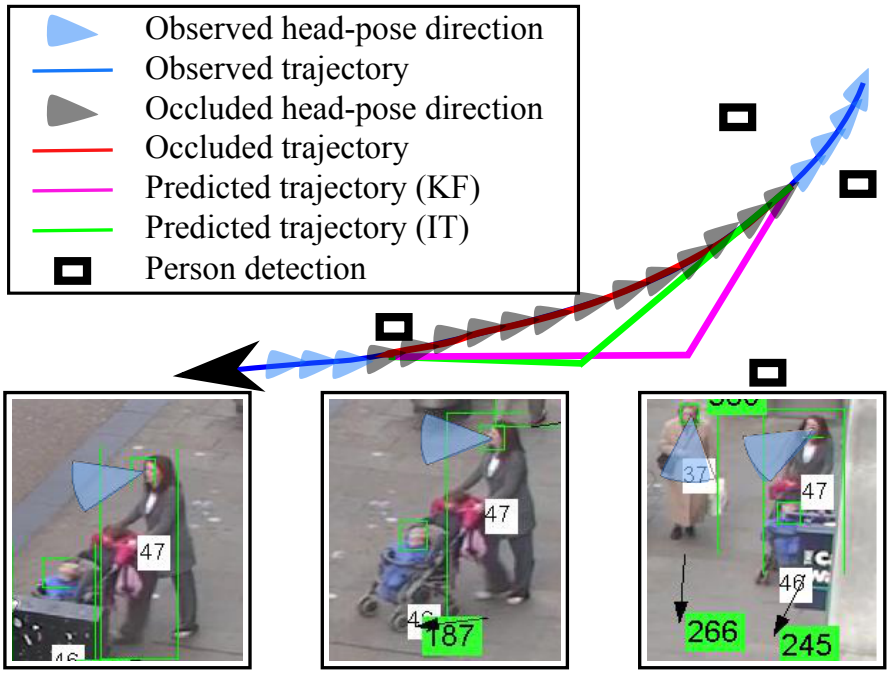



FIGURE 5.2: (*Top*) A real person trajectory/head pose behaviour and predicted trajectory using a Kalman Filter (KF) and our intentional tracker (IT). Tracking failures can lead to target data association errors. (*Bottom*) Frames from the Benfold dataset [116] showing pedestrian head-pose.

We now show how to integrate head pose information into a tracker. Note that although the algorithm is applied to pedestrian tracking our approach remains generic and different intentional priors could be used (e.g. car indicator). As a basis for our tracker we use the KF [129] due to its clear assumptions, wide spread use and efficiency.

We do tracking by detection. The input to the tracker can be from any pedestrian or head detection algorithm. For the head pose we use the Deep Belief network and CNN outputs.

### 5.1.1 Kalman Filter preliminaries

For brevity we only highlight pertinent aspects of the KF (for a thorough introduction see [130, 131]). The KF estimates the state $x \in \Re^n$ of a discrete-time controlled process governed by the linear equation $x_t = F_t x_{t-1} + B u_{t-1} + w_{t-1}$ with measurements $z_t = H x_t + v_t$ (where $t$ indicate time).

We represent the position and velocity of a target by the state vector $x_t = [pos^x, pos^y, \dot{x}, \dot{y}]^T$, where $\dot{x}$ and $\dot{y}$ represent the target's velocity with respect to its position.

$w_t$ and $v_t$ are the process and measurement noise (respectively) and are assumed to be independent and normally distributed with zero mean and covariance $Q_t$ and $R_t$ (respectively). $F_t$ relates the state of the process at $t-1$ to $t$, $B$ is the process control input model, $u_{t-1}$ is the control vector (set to 1 in the experiments) and H is the observation matrix:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{5.1}$$

That is, we measure target position but not velocity, where a measurement $z_t$ consists of the tuple $[pos^x, pos^y]_t$ and $x_0$ is initialised as: $x_0 = (H \times z_0)^T$.

### 5.1.2 Integrating intentional priors

We fuse intentional priors into the KF, firstly, by calculating the strength of the prior, denoted $\hat{s}_t$, using the absolute magnitude of the deviations for the last 10 time steps (arbitrarily chosen). This allows $\hat{s}_t$ to combine both the magnitude and persistence of the prior signal. Rather than using the raw angles, we eliminate small fluctuations in deviation/detection inaccuracies by using a binning procedure to partition the velocity and head pose into 8 bins (numerically numbered 1:8). Each bin represents a 45° sector (see Fig. 4.1). This procedure allows a smoothed estimate of the head-pose deviation signal to be generated. The signal strength at time $t$ is then calculated as follows (where $\theta_k^g$ is the head pose direction and $\theta_k^v$ is the direction of travel):

$$\hat{s}_t = |\sum_{k=t-10}^{t} Bin(\theta_k^g) - Bin(\theta_k^v)| \tag{5.2}$$

The tracker does not deal with raw headpose angles. Instead, we bin the angle into 8 discrete bins of 45° each. Hence the Bin operator collapses the raw angle into the bin center angles, namely $0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°$.

Next, we weight the influence of the prior. Intuitively, the weight ($\alpha_t$) should increase in line with the strength of the prior $\hat{s}_t$. A sigmoid function applied to $\hat{s}_t$ is a simple and effective way to achieve this. The sigmoid is parametrised by $\rho$ and $\tau$ and could be optimised for the scene to reflect the reliability of the prior, where $\rho$ adjusts the rate at which the function moves from zero to one and $\tau$ adjusts the 'base-weight' (weight given for zero strength). Rather than optimising for any particular scene, we use values for $\rho$ and $\tau$ that were empirically derived in [132] .

$$\alpha_t = (1 + exp(-\rho(\hat{s}_t - \tau)))^{-1} \tag{5.3}$$

Having determined $\alpha_t$, the transition model ($F_t$) is adjusted to reduce the influence of the target's previous motion. Denote $F_{t-1}$ as the motion model at time $t-1$ and $\gamma_t = 1 - \alpha_t$. The motion model is then updated as follows:

$$F_t = \begin{bmatrix} 1 & 0 & \gamma_t & 0 \\ 0 & 1 & 0 & \gamma_t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.4}$$

This has the effect of reducing the influence of $\dot{x}$ and $\dot{y}$ by a factor of $\gamma_t$ during the prediction step of the algorithm. The influence of the intentional prior is asserted using the control matrix $B_t$:

$$B_t = [\alpha_t dx, \alpha_t dy, \alpha_t dx, \alpha_t dy]^T \tag{5.5}$$

$$dx = d_t cos(\theta_p), dy = d_t sin(\theta_p) \tag{5.6}$$

Where $d_t$ is the geometric distance travelled by the target between $t-1:t$ and $\theta_p$ is the predicted travel direction based on head pose angle $\theta^d_{t-1}$. Two approaches could be used for calculating $d_t$: It could be estimated from $[\dot{x}_{t-1}, \dot{y}_{t-1}]$, which is an estimate of the target's velocity given observations $z_{0:t}$. Alternatively, a smoothed velocity could be calculated from $[pos^x_{t-k:t-1}, pos^y_{t-k:t-1}]$, where $2 \leq k \leq t$. In practice the second approach was found to give better performance using empirically derived $k = 5$.


FIGURE 5.3: The benefit of headpose as a prior is clearly illustrated when no prior tracking information is available. The Kalman filter output is shown in red and the intentional tracker output is shown in green. We initialize the tracker with very few frames and let the trackers evolve without further detection. (a) The person does not cross the road and his headpose at the instant of exiting the door is very indicative. (b) Similarly for people who want to cross the road, the head pose information is again very indicative of their intention. There is a region of occlusion that is shown in orange. The trajectories qualitatively show the benefit of the intentional tracker.

Having finally defined all of the components required to generate $F_t$, the remainder of the KF algorithm remains the same. Predictions are now based on a target's previous motion (with weight $\gamma_t$) and the intentional prior (with weight $\alpha_t$).

Furthermore, the instantaneous head pose prior can be used to initialize tracking where no prior tracking information is available. This can be used to approximately predict pedestrian intent with a few time steps. Figure 5.3 shows this scenario qualitatively. It can be clearly seen that the estimated head pose for people coming out of the door near the zebra crossing can be very informative in predicting their intended action.

### 5.1.3 Experiments

We compare performance of our tracker against the standard KF (by which we mean having no head-pose information) using the Benfold [116] and Caviar [133] video datasets.

We report the cumulative log likelihood (CLL) as our evaluation metric for direct comparison with [8]. Since we compare the improvements under occlusion,

TABLE 5.1: Percentage improvement (reduction) in mean squared error (MSE) during occlusion for 7 trajectories.

| Traj. No. | 7 | 8 | 9 | 10 | 11 | 15 | 22 |
|---|---|---|---|---|---|---|---|
| % Imp. | 64.0 | 75.5 | 84.5 | 12.9 | 73.2 | 62.0 | 68.3 |

it is not possible to compare with ground truth since no ground truth exist under occlusion. Hence we use Cumulative Log-Likelihood to compare the relative improvements. CLL is based on the measurement innovation and is defined as $CLL_{KF} = \sum_{k=1}^{T} LL_k^{KF}$ and $CLL_{IT} = \sum_{k=1}^{T} LL_k^{IT}$. Improvement in CLL is: $CLL_{KF}/CLL_{IT}$. CLL measures how well the innovation covariance is modelled and is a useful metric when MSE cannot be calculated. We use the same values for the parameters.

As can be seen from Figure 5.5, the intentional tracking performance is greatly improved by better headpose estimation. On the Benfold dataset we achieve a CLL median of 8.8% compared to the 5.9% achieved by their headpose estimation method. Similarly, on the Caviar dataset we achieve a CLL median of 16.02% compared to the 15.8% achieved by the competing system. It should be noted that on Caviar data, the head pose ground truth annotation based tracker gives a median CLL improvement of only 16.1% so there is very little room at the top. However in both the datasets we achieve state-of-the-art tracking performance.

Since there are only 7 examples of sudden trajectory changes in the Oxford dataset (none are occluded), we synthesised occlusions on these trajectories. Specifically, for each change in trajectory we withheld a window of observations from each tracker to occlude the change (see Fig. 1). Table 5.1 shows the improvement (i.e. reduction) in mean squared error (MSE) between the predicted and withheld pedestrian observations. A mean reduction of 62.9% was achieved across the 7 trajectories.

Figure 5.4 (a) shows performance on the video datasets when using annotated detections. Figure 5.4 (b) shows performance on the video datasets when using annotated detections. These consisted of person head-pose for the Intentional Tracker and body bounding box for the standard KF. Our approach out performs the standard KF under all conditions.

As can be seen from Figure 5.5, the intentional tracking performance is greatly improved by better headpose estimation. On the Benfold dataset we achieve a

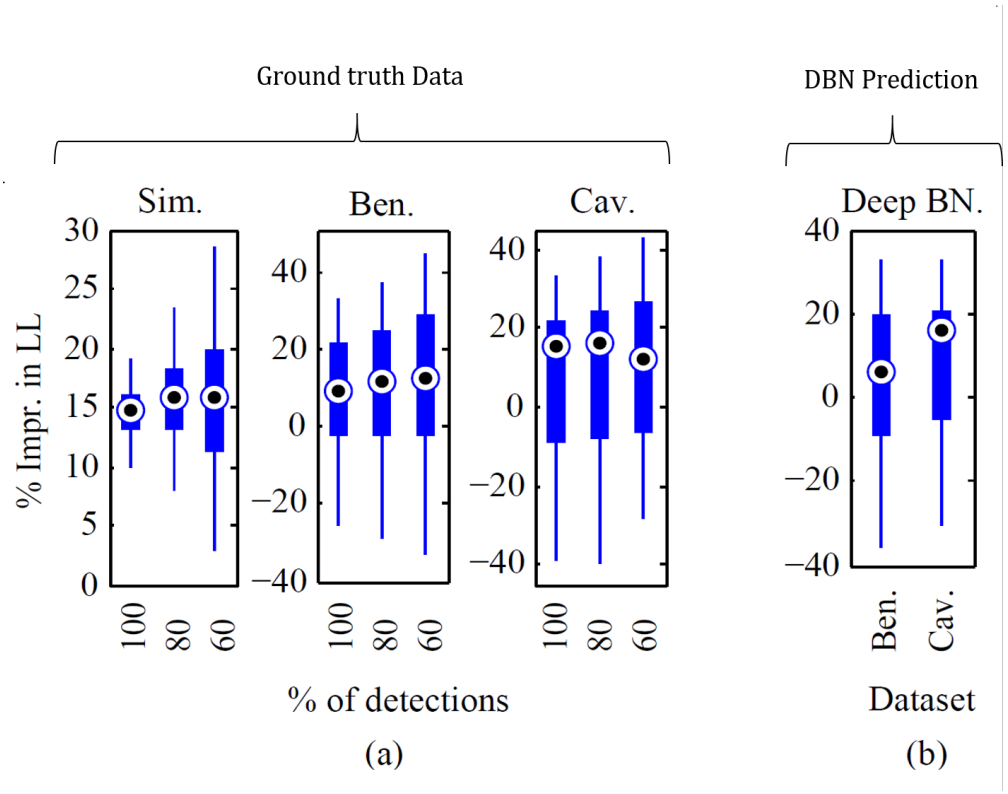



FIGURE 5.4: Improvement in Cumulative Log Likelihood (LL) by our intentional tracker vs. a standard KF. (a) Using the simulated, Benfold, & Caviar datasets under three head/body detection rates, 100%,80%,60% & ground truth head-pose. The simulated data is generated with random noise as described in [8] (b) Using head-pose classifications from our deep belief network (Deep BN) on the Benfold and Caviar datasets.

CLL median of 8.8% compared to the 5.9% achieved by their headpose estimation method. Similarly, on the Caviar dataset we achieve a CLL median of 16.02% compared to the 15.8% achieved by the competing system. It should be noted that on Caviar data, the head pose ground truth annotation based tracker gives a median CLL improvement of only 16.1% so there is very little room at the top. However in both the datasets we achieve state-of-the-art tracking performance.

## 5.2 Exploiting Headpose as a Social Signal

We use our robust head-pose estimation technique to further infer meta information regarding human centric scene understanding i.e. we wish to know what people are looking at in the real world, not merely the image plane. To this end we first define a "human attention metric" based on the regression output.

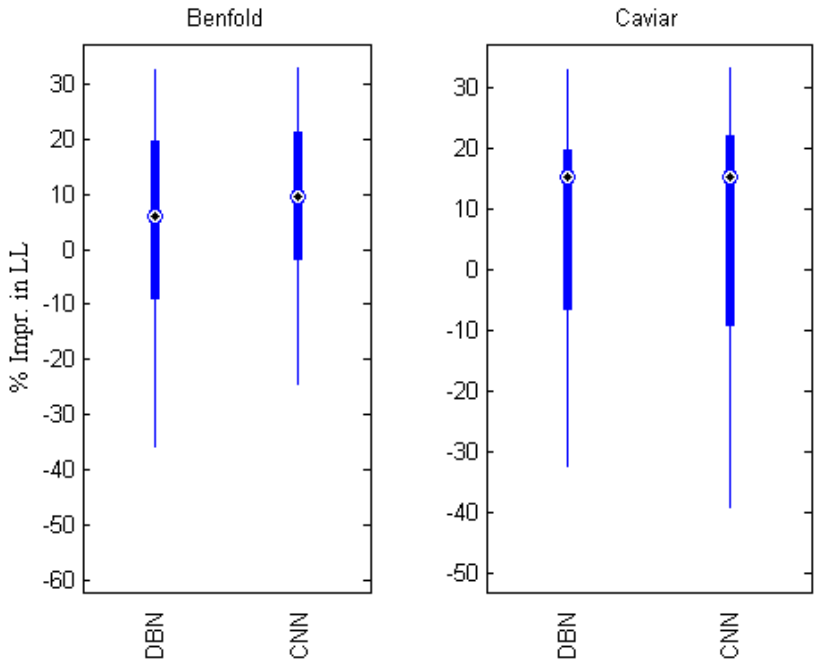



FIGURE 5.5: Comparative improvement of our headpose estimation based intentional tracking vs the method of [8]

### 5.2.1 Probabilistic Attention and Interaction Metrics

While pure head pose angle is important, we note that it carries little meaning by itself if there is no object at which the person is gazing. If we model the head-pose as a spread of attention with a mean direction and a uncertainty spread that depends upon regressor confidence that is computed by taking the variance of the classifier output (we also compute the 360°classification result along with the regression output), along with the inherent uncertainty due to not tracking the eye, we can gain a lot more useful information. Our aim is to achieve gaze estimation as a spatial probability distribution in an unified framework that can be used for both gaze estimation and interaction detection. This is distinct from approaches defined in literature. In [26] head pose is used for estimating gaze through a fixed sized disc surrounding the intersection point of the head pose ray and the object/camera plane. This approach does not incorporate the confidence of the head pose estimate to peak or diffuse the gaze estimate that our technique proposes. Whereas the LAEO system [28], while useful for interaction detection, lacks the ability to project the headpose estimate into gaze estimate. Our proposed approach, the Attention Metric (AM) solves both these problems in a unified fashion.

To define the field of attention given the Regressor output of the yaw ($\theta(t)$) and pitch ($\phi(t)$) head angles and their corresponding variances ($\sigma_1(t), \sigma_2(t)$) for each frame, we turn to the field of directional statistics. We define a unit 2-D spherical probability distribution manifold in the 3-D space using the Von Mises–Fisher distribution [134] . This distribution is analogous to a 2-D normal distribution but wrapped around a 2 dimensional unit sphere in $\mathbb{R}^{\mathbf{3}}$. In general for a $(p-1)$ dimensional sphere in $\mathbb{R}^{\mathbf{p}}$ the von Misses-Fisher distribution for the p-dimensional unit vector $\mathbf{x}$ is defined as

$$f_p\left(\mathbf{x};\mu,\eta\right) = C_p\left(\eta\right)\exp\left(\eta\mu^T\mathbf{x}\right) \tag{5.7}$$

where $\eta \geq 0$ is the concentration factor (inversely proportional to the variance $\sigma$) and $\|\mu\| = 1$ is the unit vector in the direction of the mean and $C_p\left(\eta\right)$ is the normalization factor defined as

$$C_p\left(\eta\right) = \frac{\eta^{p/2-1}}{\left(2\pi\right)^{p/2} \times J_{p/2-1}\left(\eta\right)} \tag{5.8}$$

where $J_v$ denotes the modified Bessel function of the first kind and order $v$. In our case of $\mathbb{R}^{\mathbf{3}}$ or $p = 3$ It reduces to

$$C_3\left(\eta\right) = \frac{\eta}{4\pi\sinh\left(\eta\right)} \tag{5.9}$$

Figure 5.6 shows the Von Mises-Fisher distribution for various $\eta$.

In our particular case we compute the mean direction unit vector $\mu$ from the yaw and pitch angles in spherical coordinates, and also the concentration factor $\eta$ assuming isotropic variance in both yaw and pitch angles as

$$\mu = \frac{1}{\sqrt{1+\theta^2+\phi^2}}\begin{bmatrix}1\\ \theta\\ \phi\end{bmatrix}, \quad \eta = \frac{1}{\sqrt{\sigma'^2_1+\sigma'^2_2}} \tag{5.10}$$

The $\sigma'_1$ and $\sigma'_2$ are the sum of the regressor variance ($\sigma$) and the inherent mean uncertainty (E- constant due to no eye tracking) in standard deviation units.

In case one needs to preserve anisotropic variances in both directions one can use the Kent distribution [134] which preserves those properties. However from our experience, we decided not to use it (keeping in mind its higher computational

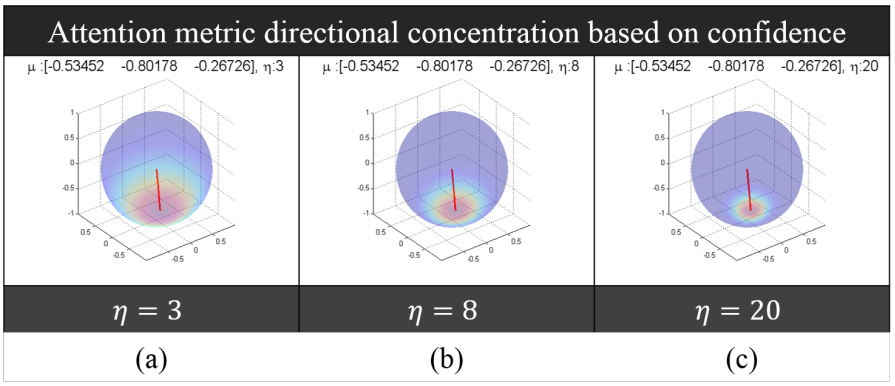



FIGURE 5.6: The Von-Misses Fisher distribution visualized on a unit sphere. The mean direction $\mu$ is represented by the red line and the factor $\eta$ represents the concentration of the distribution.

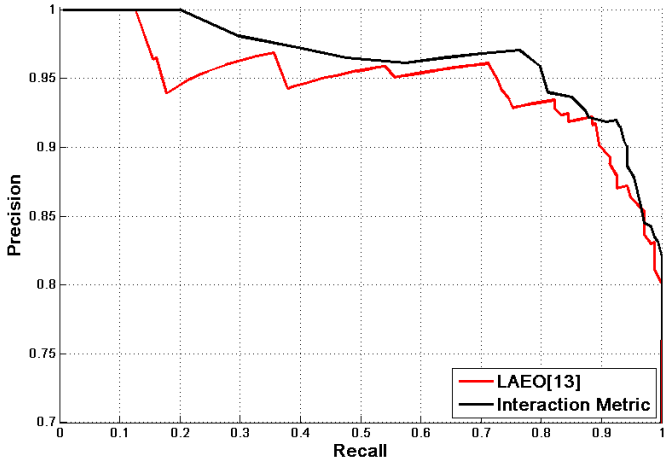



FIGURE 5.7: The precision recall curve comparing our Attention metric to the LAEO metric on our dataset

complexity). So our final attention metric for person i at time t is defined as:

$$AM_i(\mathbf{x}, \mu_i(t), \eta_i) = \frac{\eta_i}{4\pi \sinh(\eta_i)} \exp\left(\eta_i \mu_i^T(t)\mathbf{x}\right) \tag{5.11}$$

To detect interaction between any two people (i,j) we multiply two attention matrices ($AM_i$ and $AM_j$) together computed at the location of the other person's head. Hence the interaction metric (IM) for a pair of people (i,j) with their corresponding head positions $\mathbf{x_i}$ and $\mathbf{x_j}$ is defined as

$$IM_{ij} = \frac{AM_i(\mathbf{x_j}) \times AM_j(\mathbf{x_i})}{\mathbf{r_{ij}^2}} \tag{5.12}$$

where $\mathbf{r_{ij}}$ is the euclidean distance between the pair of heads.

Figure 5.8 shows the output of our Interaction metric along with interaction detection on our dataset. For comparison we also show the HLYK interaction detection scheme ,i.e. looking at each other (LAEO) as reported in [28]. We also show the raw yaw and pitch angles for both persons. In both the interaction detection signals, namely IM and LAEO the interaction ground truth is plotted in red, and the IM and LAEO signals are plotted in blue. To detect interactions from IM we can simply specify a threshold above which interaction is detected. This is the only free parameter in the IM scheme. We cross validated the parameter for various values and found that setting this IM threshold to 0.32 results in highest accuracy. In contrast, LAEO requires three free parameters, the aperture of the viewing cone $\phi$ , the temporal window for smoothing $T$ and the interaction threshold $\tau$. We computed LAEO using the best reported values for these parameters from [28]. It is note worthy that IM in bound between [0,1] allowing a probabilistic interpretation of the same, whereas the LAEO signal is not bounded. From Fig. 5.7, where we show the precision-recall curves comparing both IM and LAEO, it is evident that IM outperforms LAEO consistently across all parameter choices.

We show another instance of our interaction metric in Fig. 5.9. In this instance there are two people who are interacting in the beginning (high IM signal), then one person looks away towards the camera while the other person keeps looking at the said person (low IM signal), near the middle of the sequence they interact intermittently, and finally one person walks away. Both the binary ground truth for interaction (red) and the IM signal (blue) are shown.

Apart from showing interaction metric we also show another social signal metric called windowed cross correlation (WCC henceforth) [135]. This signal measures the similarity between any pair of time series head pose signals (within some time window; leading or lagging) and can be used to detect group behaviour.

To further show our system we consider the scenarios shown in Figures 5.10 and 5.11 by using the both the interaction metric (IM) and WCC signals. In the case of Fig. 5.10, the scenario starts with two people looking with each other. During this period we see that the IM signal is indicative of the scenario. Then both the persons start walking together in the same direction. This leads to a high valued WCC signal and zero IM signal. This state of the signals persists as both of them look together into the same object of interest. Finally, one person walks away before the other causing a drop in the WCC signal, which again goes up as the

other person joins moments later. All this while the IM signal is zero or near-zero as no interaction is taking place.

In the case of Fig. 5.11, the scenario represents two people loitering in a common area until suddenly something attracts their attention and both look towards the same thing. This leads to a low WCC signal at the beginning which is followed by a high WCC signal when both of them look towards the same thing. Finally when they move apart from the scene we see a corresponding drop in the WCC signal.

**Discussion:** The IM signal was quantitatively evaluated in Fig. 5.7 and compared against the state-of-the-art metric LAEO as described in [28]. The IM signal with significantly less number of free parameters to tune (one, namely the interaction threshold) outperformed LAEO in all scenarios. Subsequently from all the qualitative scenarios described in Figures 5.8, 5.9, 5.10, and 5.11, we see that both the IM and WCC signals are intuitive, and both provide key evidences that can contribute towards higher level behaviour inference in social signal processing.

## 5.3 Discussion

In this chapter we applied the robust headpose signal estimation to understanding human behaviour. First our intentional tracker is shown to be superior to standard Kalman filter in presence of occlusion. Then we established the Attention Metric and the Interaction Metric that explicitly models the output confidence of the regressor/classifier. We reported state-of-the-art results on standard datasets for detecting when people are looking at each other. We also qualitatively evaluated all the three metrics, including the Windowed Cross Correlation (WCC), on three scenarios and showed the validity of the assumptions. We have contributed these three strong baseline metrics to the broader literature of Social Signal Processing where other systems can be built on top.

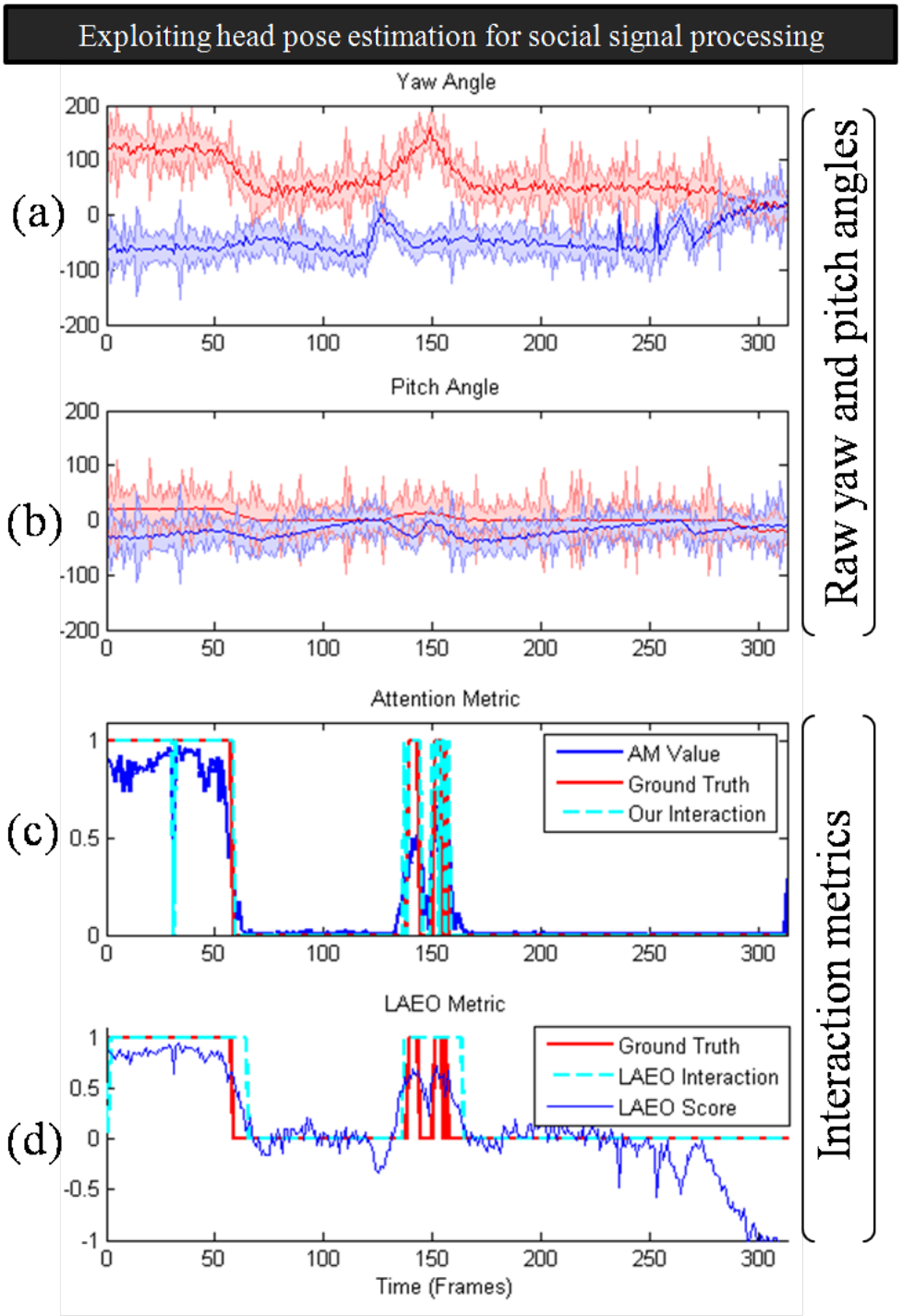



FIGURE 5.8: In this figure we show the use of the head pose angles with the interaction metrics LAEO [28] and AM to do interaction detection. Subfigure (a) and (b) show the yaw and the pitch angles along with their 95% confidence intervals, of two heads in a sequence of two people interacting. Subfigure (c) shows the output of our our Interaction Metric (IM, in blue) and interaction detection (dotted cyan) and subfigure (d) compares the LAEO [28] metric (blue) and its corresponding interaction detection (dotted cyan). The ground truth for interaction is shown for reference in red in both (c) and (d) and the signal is binary (interaction is either happening, or not). IM clearly outperforms LAEO.

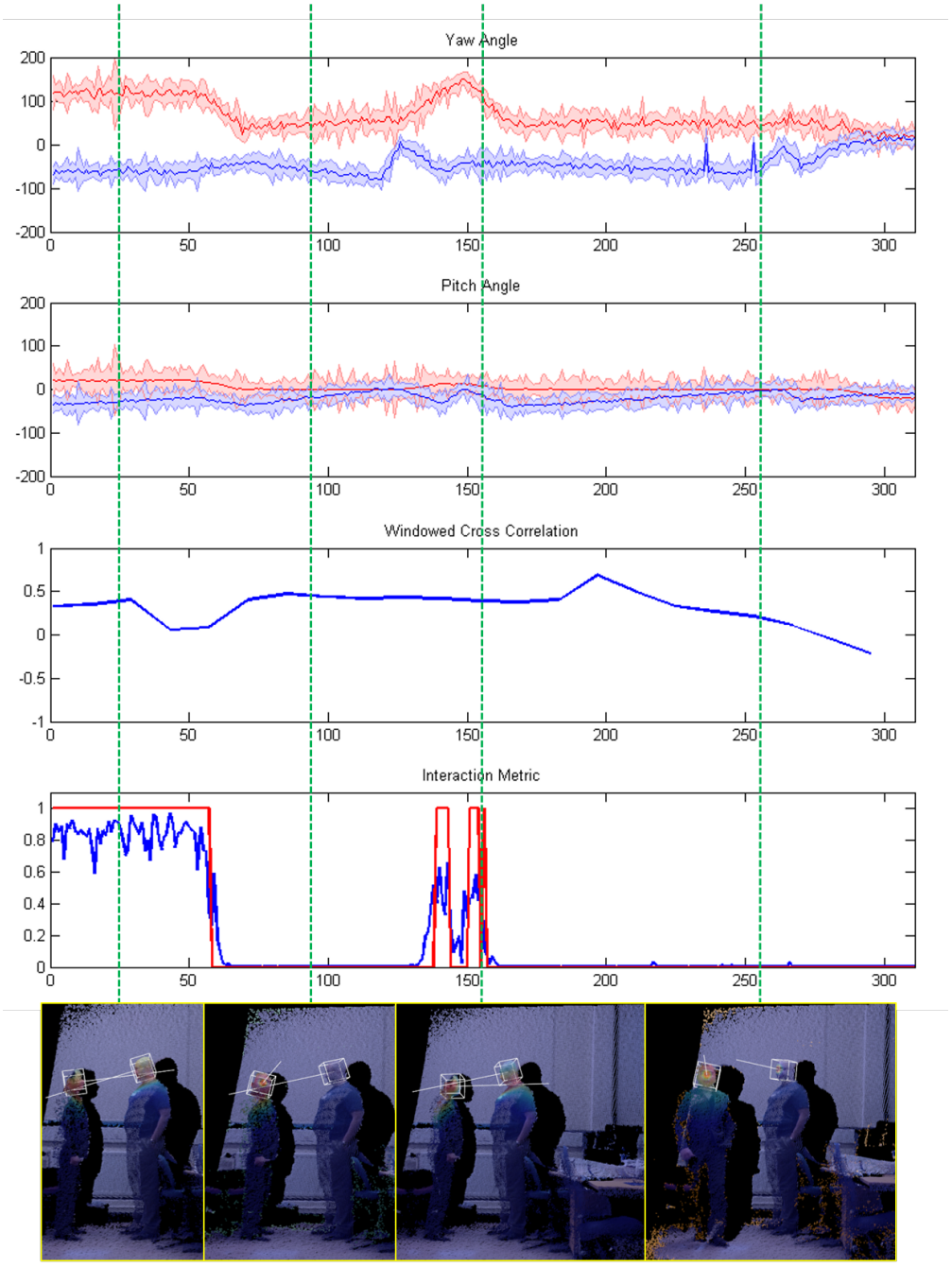



FIGURE 5.9: We show the interaction and WCC head pose signals. The binary ground truth for interaction is shown in red. The raw head pose signals are same as 5.8 (a) and (b). The scenario can be described with the four snap shots as follows: (1) Two people are talking facing each other and from (b) the IM can be observed to be high while from (a) the WCC is not observed high. (2) One person looks away towards the direction of the camera which followed by a drastic fall in the IM in (b) while the WCC in (a) falls while the two heads behave differently and stabilizes. (3) The person looks back intermittently and we see the corresponding change in IM. (4) Finally the person walks away with the other person looking at the same place. This makes the WCC falling drastically. The peak WCC is achieved around frame 200 when both of them look at the general direction of the camera.

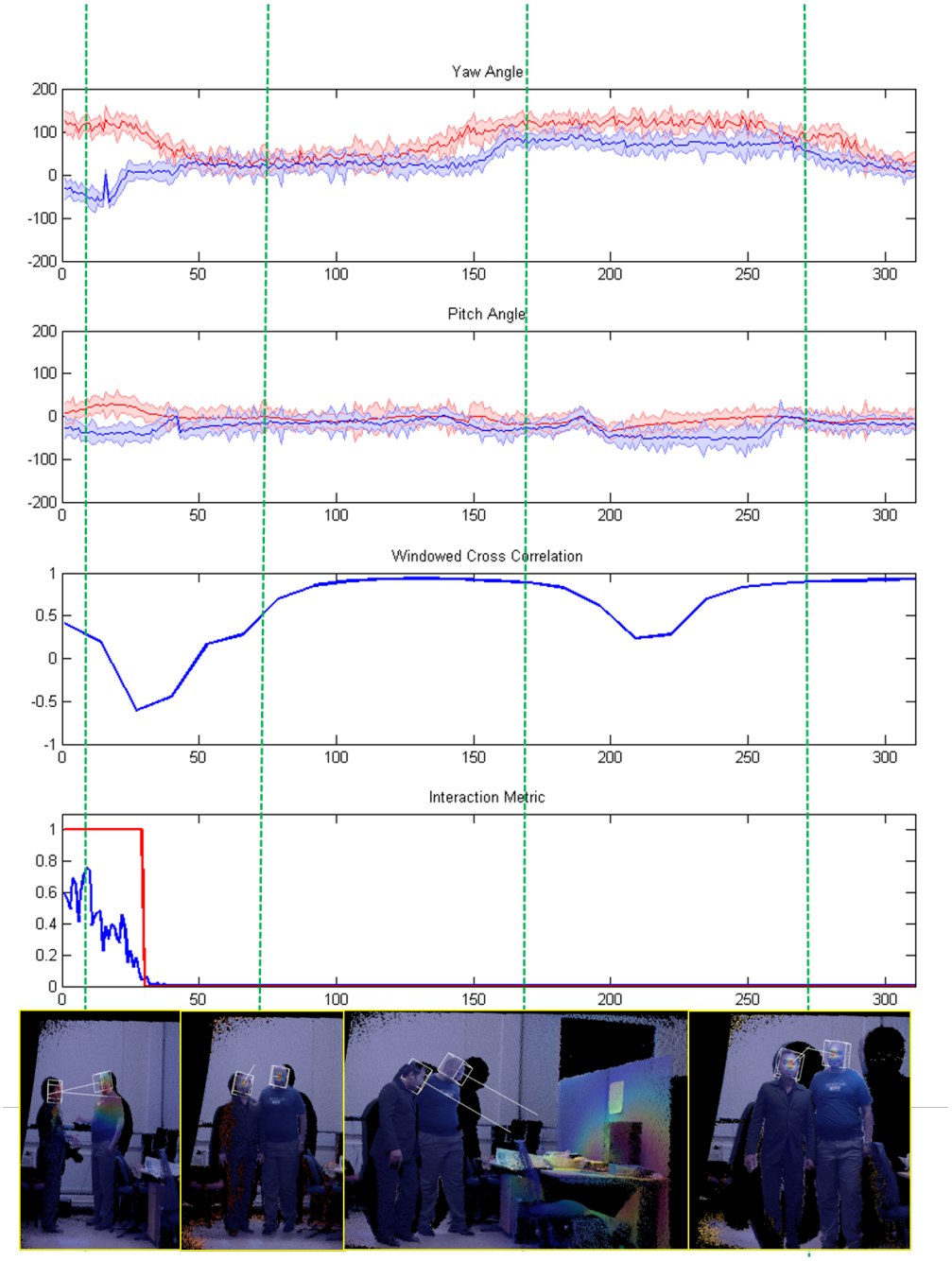



FIGURE 5.10: In this scenario two people are interacting as can be seen from (1). This results in the corresponding IM and WCC signals in (b) and (a) respectively. Then in (2) they start walking together in the same direction facing the camera. This makes the WCC signal go up. *This shows the usefulness of WCC in detecting social mimicry.* The WCC signal stays high when in (3) they look at the same object of interest together. Finally in (4) they walk away together looking towards the camera. The dip in the WCC signal near frame 220 is caused when one person walks away before the other.

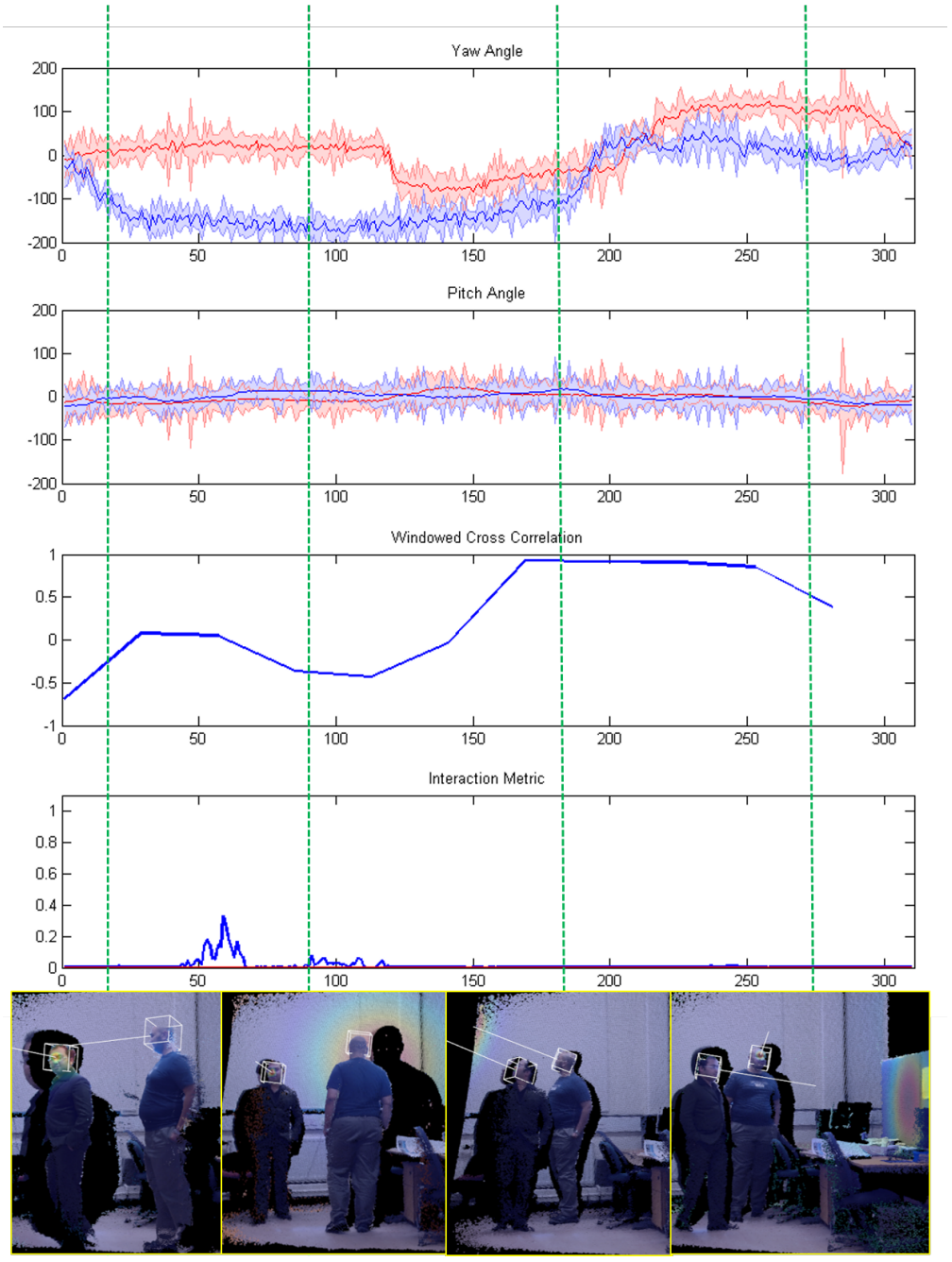



FIGURE 5.11: In this scenario the two people are not behaving similarly at the beginning and are looking at different things at different times. Finally in (3) they are attracted by the same thing on the wall on the left and look at it together. This makes their head pose signal to become highly correlated as can be seen from (a). Finally they walk away their separate ways and we see a drop in the WCC signal.

# Chapter 6

# Coral Image Segmentation: AN Application of ALICE Loss

*We digress to an application of the ALICE loss that was developed for the end-to-end detection and classification in Chapter 5, to dense classification of underwater coral reefs imagery. The objective of this work is to solve the challenging task of recognizing and segmenting underwater coral imagery in the wild with sparse point-based ground truth labelling. To achieve this, we propose an integrated Fully Convolutional Neural Network (FCNN) and Fully-Connected Conditional Random Field (CRF) based classification and segmentation algorithm. Our major contributions lie in four major areas. First, we show that multi-scale crop based training is useful in learning of the initial weights in the canonical one class classification problem. Second, we propose a Adaptive Localizing Infogain Cross-Entropy Loss (ALICE) for training the FCNN on sparse labels with class imbalance and establish its significance empirically. Third we show that by artificially enhancing the point labels to small regions based on class distance transform, we can improve the classification accuracy further. Fourth, we improve the segmentation results using fully connected CRFs by using a bilateral message passing prior. Finally, we improve upon state-of-the-art results on all publicly available datasets by a significant margin. This work has been presented in the 6th International Symposium on Deep-Sea Corals [19].*

Coral reefs are Physical structures built by the actions of many tiny coral animals. They are most diverse underwater ecosystems widely used and economically valuable to millions of people as important sources of food and income, serve as living home for the great amount of fish species, attract divers, snorkelers and underwater tourists, generate the sand on beaches, and protect shorelines from damages by storms [136].

However, their existence vulnerable as result of man-made threats. According to recent study in [136] more than 60% of the world's reefs are under threat. Overfishing and destructive fishing are the most prevalent practices that affect more than 55% of the reef in the world. When the effects of recent thermal stress and coral bleaching are combined with the other threats, the effect of thermal stress is also quite high. The estimate of threat to reefs across the region increases to more than 90%, with the percent of reefs rated at high or very high increasing to nearly 55%.

Previously coral reef restoration process involves extreme conditions for volunteer SCUBA divers, who transplant loose fragments from seabed back onto the larger reef framework. However, limitation as a result of low temperature arises as the height increases beyond some depth level. The Coralbot project is a recently proposed idea to autonomously repair deep-sea coral reefs.

A supervised automated labelling tool has become necessary that can be integrated in AUVs to help detect corals. The aim of this project is to develop the machine vision algorithms to help a Coralbot to locate a coral reef and a chunk of coral on the seabed and prompt the Coralbot to pick it up. This involves mainly classifying different types of coral types. The technical challenges are principally due to the potential lack of clarity of the water, platform stabilization, spurious artefact (rocks, fish, crabs etc) and lack of training data, very high intra-class variability.

We develop a deep CNN coral classifier that can be classify corals for the specified objective with sufficient accuracy. The proposed method is compared with state of the art methods on recently proposed benchmark coral dataset, the Moorea labelled corals (MLC) dataset, and shows relative improvement on overall accuracy. The consistency of the proposed method is evaluated on smaller dataset prepared for this purpose, the Atlantic Deep-Sea dataset (ADS) dataset.

Next, we present the related works, in which we summarise the main theories and current researches that explain the classification of coral reef. Then, we illustrate

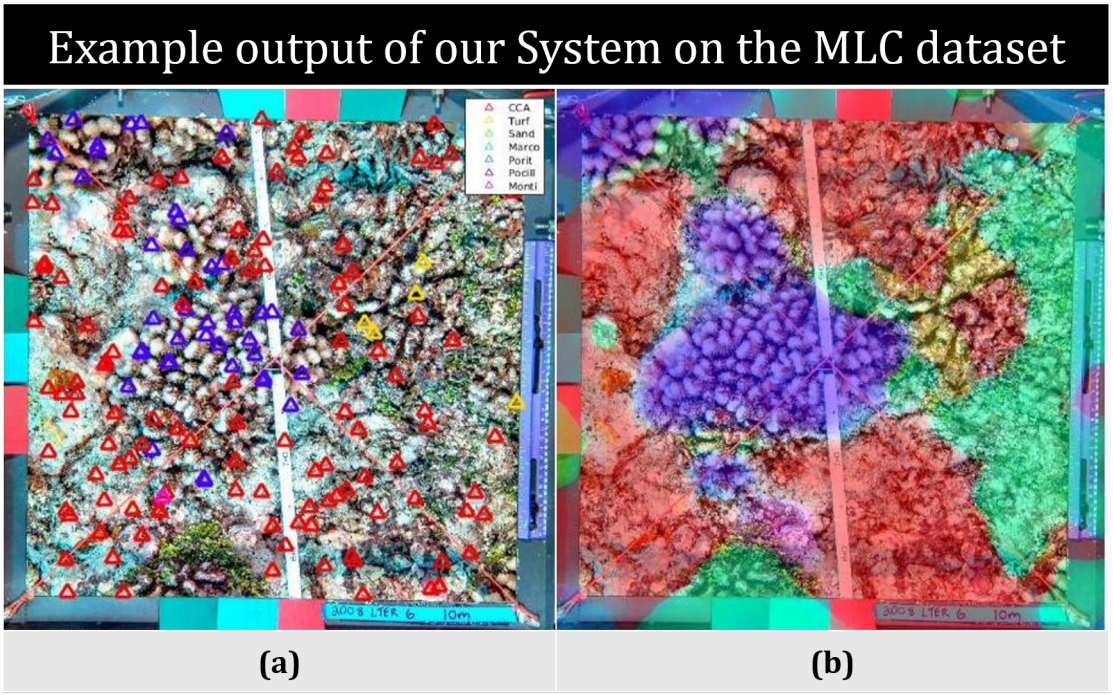



FIGURE 6.1: Illustrative output of our system on the MLC Dataset. (a) Shows the sparse point based ground truths. (b) Shows the pixel-wise classification

the development of the proposed methodology and the results of the of coral classification. Finally, we present conclusions.

## 6.1 Related Work

Our approach draws on recent successes of deep nets for image classification [33, 127] and fine-tuning techniques for fast and improved classification specially for small datasets [137, 138]. The following subsection discuss related works on coral classification and relevant works on object recognition using deep CNNs that are used as input for this work.

### 6.1.1 Automated Coral Classification

There are couple of works on literature that deal with coral classification. One of those, the work by Marcos *et al.* [139] use a feed-forward back-propagation neural network to classify close-up images of coral reef with success rate of 86.5%

on test sets using Color and texture features. This performance might vary under different Coral datasets.

A Mehta *et al.* [140] present a method to classify coral reef images based on their textural appearance using support vector machines. The advantages of this method are its simplicity and uses a small number of images for classification though it performs poorly for noisy and under non-uniform illumination.

Stokes and Deane [141] describe an automated computer algorithm for the classification of coral reef benthic organisms and substrates. Both texture and colour features are extracted using discrete cosine transform and normalized colour space respectively and probability density weighted mean distance is used for classification. Although good accuracy are obtained with colour correction methods, achieving consistence result is challenging in underwater imaging.

Beijbom *et al.* [142] present a novel frame work to classify coral reefs using both colour and texture features. They used maximum response filter bank to extract features and SVM with radial basis kernel for classification and achieve 83.1% on his nine class MLC dataset.

The work by Tusa *et al.* [143] describes the development of a vision system for coral detections based on supervised machine learning. Gabor wavelet filters are used to extract texture features and a good accuracy is obtained using decision tree algorithms on their dataset. The over-all accuracy obtained is not high enough and the result is yet to be verified on other datasets.

The work of Shihavuddin *et al.* [144] discusses classification benthic coral reef using different machine learning methods. They used completed local binary pattern, gray level co-occurrence matrix, Gabor filter response, and opponent angle and hue channel colour histograms to extract feature descriptors. For classification Knearest neighbor, neural network, support vector machine and density weighted mean distance are investigated.The performance is evaluated on seven datasets including three texture datasets. An accuracy of as high as 85.5% is obtained on Moorea labelled corals (MLC) 2008, one third of the MLC dataset, dataset.

These works have all used hand-crafted features for coral classification. We proposed a deep learning method that extract and learn useful features automatically without concerning on the preprocessing of raw input images. Another limitation of the previous works lie in their requirement to extract fixed size crops (sometime

discrete multiple sizes) around the labelled points for classification. This limits the applicability to only those images where points of interests have already been marked. We alleviate this problem by treating it as a segmentation problem where every pixel will be given a corresponding label. We use the terms dense classification, classification with localization and semantic segmentation interchangeably in this paper. Our approach has the added benefit of allowing the computation of relative coral area coverages for free. This is a very important statistic in marine biology for coral reef surveys and health monitoring.

## 6.2 CNN Architecture for Patch Training

As result of success in different benchmarks there have been a lot of changes in architecture compared to the first proposed deep CNN by Yann LeCun [145]. More recently developed architecture are becoming in general deeper and wider, showing state of the art performance in object recognition challenges [33, 34, 127, 146]. The power of the CNNs lie in their ability for end to end learning of multiple layers of non-linear transform on the data from the data itself in a supervised manner. The operations of Convolutional Neural Networks can be follows

1. **Convolutional Layer:** A three dimensional feature map at layer $l$ consists of $m_1^{(l)}$ feature maps of size $m_2^{(l)} \times m_3^{(l)}$. The $i^{th}$ feature map in layer $l$, is denoted by $Y_i^{(l)}$. For a given bias matrix $B_i^{(l)}$ and kernel $K_{i;j}^{(l)}$ of size $(2h_1^{(l)}+1) \times (2h_2^{(l)}+1)$, the output feature map at layer $Y_i^{(l)}$ at position $(r, s)$ is computed as

$$(y_i^l)_{r,s} = (B_i^l)_{r,s} + \sum_{j=1}^{m_1^{l-1}} (K_{i,j}^l * y_j^{l-1})_{r,s} \tag{6.1}$$

$$= (B_i^l)_{r,s} + \sum_{j=1}^{m_1^{l-1}} \sum_{u=-h_1^l}^{h_1^l} \sum_{v=-h_2^l}^{h_2^l} (K_{i,j}^l)_{r,s} (y_j^{l-1})_{r+u,s+v} \tag{6.2}$$

   The trainable parameters are stored in filter $K_{i,j}^l$ and bias matrix $B_i^l$.

2. **Pooling:** Pooling layers downsample the feature maps by selecting maximum or average values across spatial dimensions. Successive layers of pooling makes the features more translational invariance which is a desired property for object recognition. Pooling operations may be overlapping or non-overlapping. There

can be many types of pooling schemes. However the most popular ones are max or average pooling. In our work we have used max pooling without overlap.

3. **Activation function:** Non-linearities introduced at each stage after convolution makes the learned features more robust. Rectified linear unit (ReLU) is applied to active outputs of the convolution layers. Apart from rectified linear unit, there are other non-linearities like tanh, sigmoid etc.

4. **Initialization:** Recently it has been found out that good weight initialization can mitigate the problem of the training getting stuck at local minima. During the resurgence of neural networks in the last decade, good weight initialization using unsupervised pre-training in the form of Restricted Boltzmann Machines or Auto-encoders were proposed as a solution. However, as can be seen from recent work [147], for large supervised datasets good weight initialisation by fixing the limits of the random numbers are just as good. This scheme called the Xavier weight initialisation scheme can be defined as follows

$$w_{ij} = u\left[\frac{\sqrt{6}}{\sqrt{n_j + n_{j+1}}}, \frac{\sqrt{6}}{\sqrt{n_j + n_{j+1}}}\right] \tag{6.3}$$

where $\mathrm{u}\big[a, a\big]$ as the uniform distribution in the interval $(-a, a)$. $n_j$ is the size of the previous layer output and $n_{j+1}$ is the number of output of the current layer.

5. **Optimization:** Mini batch gradient descent with momentum update is used to computes the update based on small batch of images. Dropout is also used as a regulariser to make the network robust to overfitting.

6. ***Learning rate:*** Step and power decay methods are considered to decrease the initial base learning rate for this work.

### 6.2.1 Fine-tuning

Adapting an already trained model(most of the time on Imagenet as it is quite general), for different unrelated tasks has been shown to be quite effective [137, 138]. This is especially useful if there is a scarcity of labelled data. Though the datasets are visually similar to ImageNet, those successful works inspired us to experiment the technique using the state of the pre-trained ImageNet models.

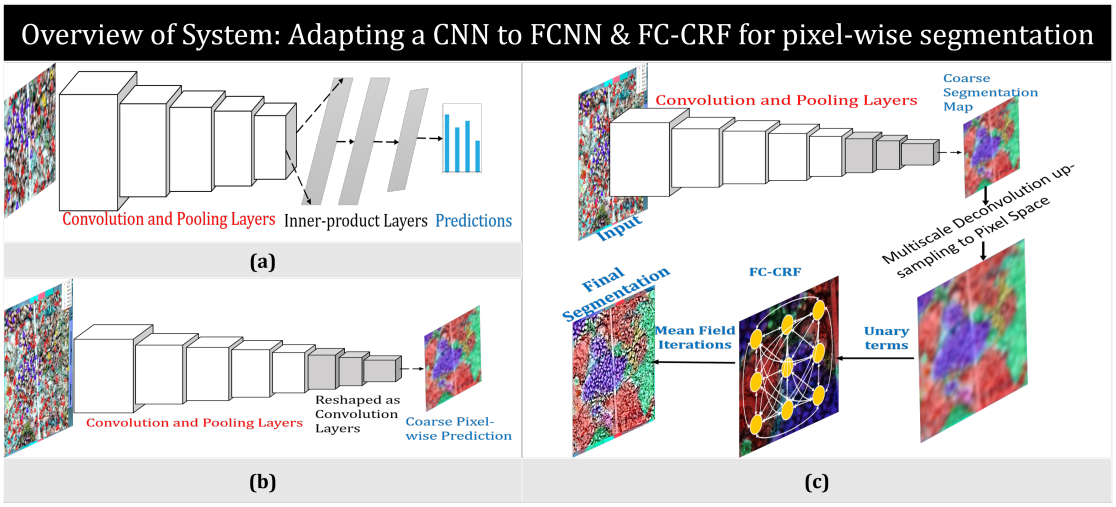



FIGURE 6.2: Adaptation of CNN for pixel-wise classification. (a) Standard VGG-16 net trained in patchwise training based on . (b) Conversion of (a) into a fully convolutional architecture that outputs coarse classification map with sparse point based training. (c) Finally large field of view and low stride version of net output that is post processed with FC-CRF for final pixel wise classification.

The models selected for fine-tuning are the Googlenet [38] and Visual Geometry Group (VGG) [127] models (VGG-16) based on their exceptional performance in the Large Scale Visual Recognition Challenge 2014 (ILSVRC2014) challenge. There are few underwater images in ImageNet visually they are hardly similar to the MLC and ADS dataset, but the trained models are believed to mimic human visual cortex to some level.

## 6.3 FCNN Architecture for Semantic Segmentation

We convert the patch trained CNN architecture to fully convolutional architectures by converting the Fully-Connected (FC layers) into $1 \times 1$ convolutions as shown in [42]. This does not increase the field of view of the layers and makes the architecture resolution independent. However, as shown in [42], naively doing this does not guarantee good performance. For example, the VGG-16 architecture final layer has a stride of 32 pixels and hence the resulting feature resolution is too coarse for accurate segmentation. To overcome this we replace the convolutions in the last three convolutional layers with dilated convolutions as described in

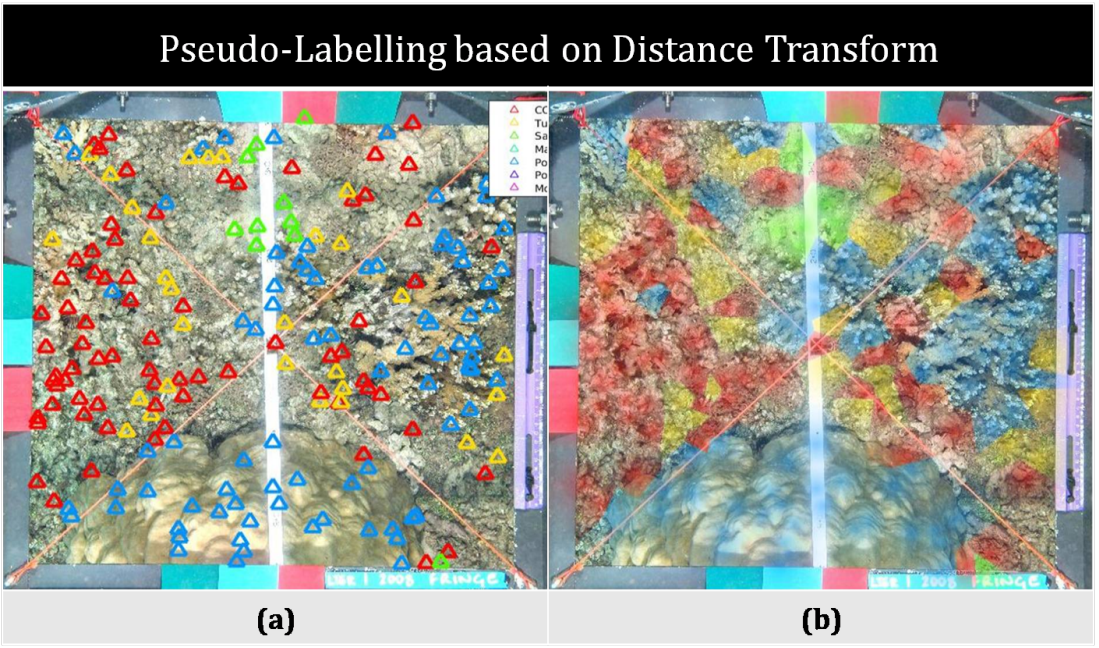



FIGURE 6.3: Pseudo-label generation based on distance transform. (a) Provided sparse point based ground truth. (b) Generated pseudo label based on distance transform with post-processing $\gamma = 5.5$ and $\tau = 0.05$

[96]. Figure 6.2 shows the architecture adaptation. This leads to a fairly dense prediction at 8 pixel stride. The network architecture is thus identical to [96].

### 6.3.1 Modified ALICE loss

We use the ALICE loss defined in Chapter 4.3.5 to tackle this problem as well. We show that ALICE loss is applicable broadly to pseudo or weakly labelled data.

The only modification we make is since our ground truth is point based we do not confine the pseudo distance transform based label within any bounding box. Instead we compute it all over the image. To achieve this we separate each of the class into its own image plane and apply distance transform [128] and then merge them back into one image based on each pixels distance from the closest class point weighted by the distance from the true point. For example if we have C classes, then we create C image planes of the same width and height of the image, one for each class. We initialise the images planes to zero value and then we place each point label annotation in the planes according to class label. Hence each image plane represents all the points that belong to that corresponding class. We then apply the distance transform to each image plane separately. We normalize the

distance transform to be in the range [0,1] by dividing it by the length of the diagonals of the image. Then we combine the C class distance maps to one class-distance map where at every pixel we choose the lowest distance from the corresponding pixels of the class maps. Hence this class-distance map has a class label and the distance value at every pixel. We vary the $\gamma$ and $\tau$ and find out optimal values for them through cross-validation during our experiments. Figure 6.3 shows the resulting pseudo labels using $\gamma = 5.5$ and $\tau = 0.05$ which were found to be optimal.

We use the distance transform instead of standard image processing based segmentation like graph-cut because of two reasons. Firstly, the labels for our problems were very sparse and hence by initialising graph-cut based on these points didn't provide good results in our experience. Secondly, we wanted to build the ALICE loss in such a way that it assigned the most weight to the actual labels and lesser weight to the pseudo-labels. This distance transform based formulation helps us preserve the relative strength of the labels unlike other segmentation approaches which would assign a hard label at every point. Any non-optimal results from these algorithms would adversely affect the training of the neural networks. By defining the ALICE loss like we did, we address both these issues.

### 6.3.2 Dense Conditional Random Field for Improvement of Segmentation

Fully connected CRFs have been shown to perform exceedingly well in fine-grained image segmentation [148]. The dense CRF model over an image is defined as follows.

$$P(\boldsymbol{x}) = \frac{1}{Z} \exp\Big( - E(\boldsymbol{x})\Big) \tag{6.4}$$

where $\boldsymbol{x}$ is the label for pixels, and $E(\boldsymbol{x})$ is the energy function. For semantic segmentation the energy function can be defined as

$$E(\boldsymbol{x}) = \sum_i \theta_i(x_i) + \sum_{ij} \theta_{ij}(x_i, x_j) \tag{6.5}$$

where the unary potential $\theta_i(x_i) = -\log P(x_i)$, $P(x_i)$ is the probability of assigning label to pixel $i$ (which is the posterior of our FCNN), while the pairwise potential is $\theta_{ij}(x_i, x_j) = \sum_{m=1}^{K} w^m \cdot k^m(\boldsymbol{f}_i, \boldsymbol{f}_j)$. The edge set is the fully connected edges between any $i$ and $j$, and each $k^m$ is a high dimensional Gaussian kernel depends

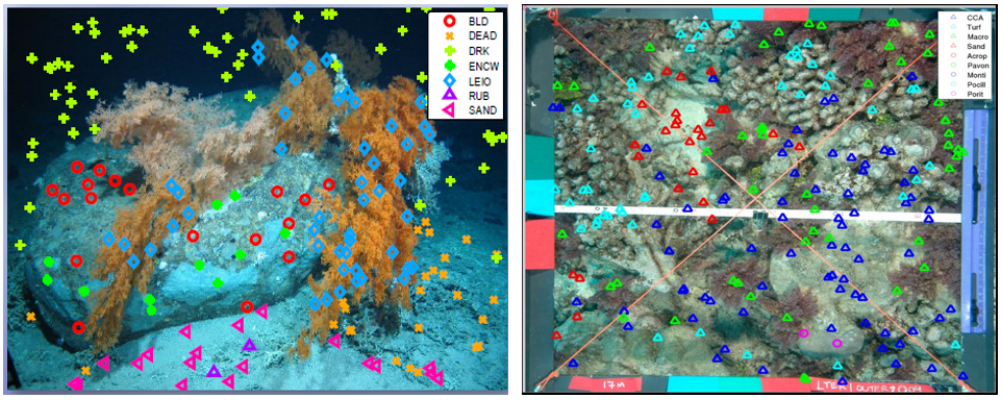



FIGURE 6.4: Sample image from ADS (right) and MLC dataset (left). The 200 points show how the labelling is done for a single image. Markers of the same colour show corals of the same class

on features computed for each pair of pixels $i$ and $j$ and is weighted by parameter $w^m$.

$$\theta_{ij}(x_i, x_j) = w^1 \exp\Big(-\frac{|p_i - p_j|^2}{2\sigma_\alpha^2} - \frac{|I_i - I_j|^2}{2\sigma_\beta^2}\Big) + w^2 \exp\Big(-\frac{|p_i - p_j|^2}{2\sigma_\gamma^2}\Big) \tag{6.6}$$

where the first kernel is a 5 dimensional Gaussian kernel that depends on both pixel positions ($p$) and pixel color ($I$). This is equivalent to bilateral filtering. For efficient evaluation of the filter in $O(n)$ time a permutohedral lattice based algorithm has been used [149]. The second kernel only depends on pixel positions and is like a standard Gaussian blur filter. The hyper parameters $\sigma_\alpha$, $\sigma_\beta$ and $\sigma_\gamma$ specify the variance of the Gaussian kernels.

The efficient inference algorithm is based on mean field approximation to the CRF distribution, which simplifies the distribution to $b(\boldsymbol{x}) = \prod_i b_i(x_i)$ and minimizes the KL-divergence $KL(b(\boldsymbol{x})\|P(\boldsymbol{x}))$.

## 6.4 Dataset

The dataset used for this work are the MLC and the Atlantic Deep sea coral dataset (ADS) dataset. The MLC dataset [142] consists 2055 high resolution images from three habitats: fringing reef, outer 10m and outer 17m, acquired in years 2008, 2009 and 2010. The seven most abundant corals are considered for this work.

The ADS dataset consists 159 images of size 2592 X 1944 collected from north Atlantic west of Scotland and Ireland in year 2012 from depth of 100-800m. This data set is prepared for this research work purpose. Similar to the MLC, 200 random points generated in each image are annotated by experts. We consider only the six most abundant corals in this dataset.

The two datasets show common problems of: mislabelling in which two very close points are annotated using different labels, labelling ambiguity, high inter-class similarity on some of their coral types and high artefacts specially in MLC dataset.

## 6.5 Data Augmentation

Taking appropriate patch size from the high resolution images in the dataset is an open problem. A trivial approach is to evaluate the performance of different patch sizes. Patch size of 64 and 200 are experimented. Patching and dataset splitting to training and validation sets are done simultaneously. 90% of each class is considered for training and the rest for validation. Because of labelled data limitation, we didn't use any separate test set. The class with few image samples are augmented and the data is resized to 256 X 256. The training is done in Caffe [124]. The central crop of size 224 X 224 are actually used for training and validation. Each training samples are also flipped. The training data mean is subtracted which is essential to centre the data so that its mean is zero for efficient learning.

Works mentioned [150, 151] describe the advantage of data augmentation to improve performance of Deep architectures and overcome overfitting[34]. We apply geometric transformations(translation, scaling, homography, flipping and rotation) to introduce class imbalance and tackle the problem of overfittig by increasing the number of minor Corals mainly. We are able to increase the training samples in the MLC dataset by factor of 3 to 5 using scaling, homography, flipping and rotation. For ADS dataset, the amount of data is increased by factor of 5 to 12 using all possible transformations.

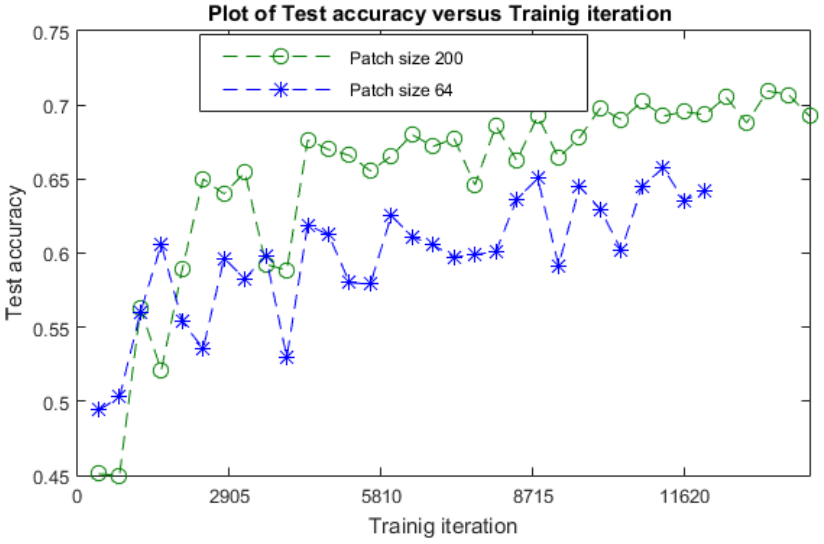



FIGURE 6.5: Effect of different Patch size on the test accuracy

| Parameter | GoogleNet | VGG16 Net |
|---|---|---|
| Training | SGD | SGD |
| Batch size | 64 | 32 |
| Starting Learning Rate($\eta$) | 0.01 | 0.01 |
| Momentum ($\alpha$) | 0.9 | 0.9 |
| $\eta$ Decay factor(no. batches) | 0.96(2000) | 0.75(2500) |
| L1 weight regularization | $10^{-5}$ | $10^{-5}$ |

TABLE 6.1: Parameters for training CNNs

## 6.6 Patch Based Training

First the vanilla classification networks (Googlenet and VGG16) are trained on the extracted patches. Two separate experiments are performed, one for patch size $64 \times 64$ and the other for patch size of $200 \times 200$. As can be seen from Figure 6.5, a patch size of $200 \times 200$ provides the best classification results and fastest convergence in training. Hence, for the training of the patch based classification experiments, this patch size has been used. The training is stopped once the test results degrade (or does not improve) for at least 2 epochs.

The patch based classification experiments we tried training both the GoogleNet and VGG16 nets from scratch, and also experimented with finetuning from their Imagenet trained models. For finetuning all the architectures, we first train using the ADS dataset and then further finetune on the MLC dataset. One of the problems of the patch based classification is that, since the field-of-view of the

architectures are large, i.e. $224 \times 224$, multiple classes of coral might be present in the patches. This leads to degraded classification performance which can be alleviated by densely localizing and classifying the images with fully convolutional neural networks. Hence we have adopted the FCNNs for better performance. This hypothesis is validated in Section 6.8 where one can see that by localizing and classifying at the same time performance does indeed improve.

Although our labels are provided as point instances, we note that during test time without classifying a patch at every single pixel, we can not produce class instance classification. This is a weakness of all the path based approaches which includes all prior work on this problem. Hence we produce a segmentation based solution. To bootstrap the segmentation network, we first train a patch classification network by pretraining on the patch based data. Then we modify the patch based network to produce segmentation and finetune on the segmentation task based on ALICE loss.

## 6.7 Classification and Localization

The trained classification models are converted to fully convolutional ones by reshaping the fully connected layers to convolutions. For the GoogleNet architecture it is a simple conversion of the final classification fully connected layers into $1 \times 1$ convolutions. For the VGG16 the last three fully connected layers (FC6, FC7 and FC8) are converted to convolutional layers with filter sizes of $7 \times 7$, $1 \times 1$, and $1 \times 1$ respectively [42]. However as shown in [152] the $7 \times 7$ filter is a computational bottleneck and by simple down-sampling the learned filters by spatial decimation to $4 \times 4$ or $3 \times 3$ does not affect performance negatively. For our experiments we converted the FC6 convolution filters to $3 \times 3$. The rest of the architecture is same as the FCN-8s architecture described in [42]. The FCNN is fine-tuned using the generated pseudo labels as described in subsection 4.3.5. The best values for the parameter $\gamma$ as described in equation 4.16 has been established by cross validation. The learning rate was set to $10^{-6}$ and reduced by $10^{-1}$ after every 5 epochs. The output posterior of the FCNNs are then used as input to the DCRF model. The number of mean-field iterations for the DCRF are fixed at 5 iterations. In our experience more iterations provide diminishing returns compared to the computational cost. We compare four models in this experiment, 1. fully convolutional

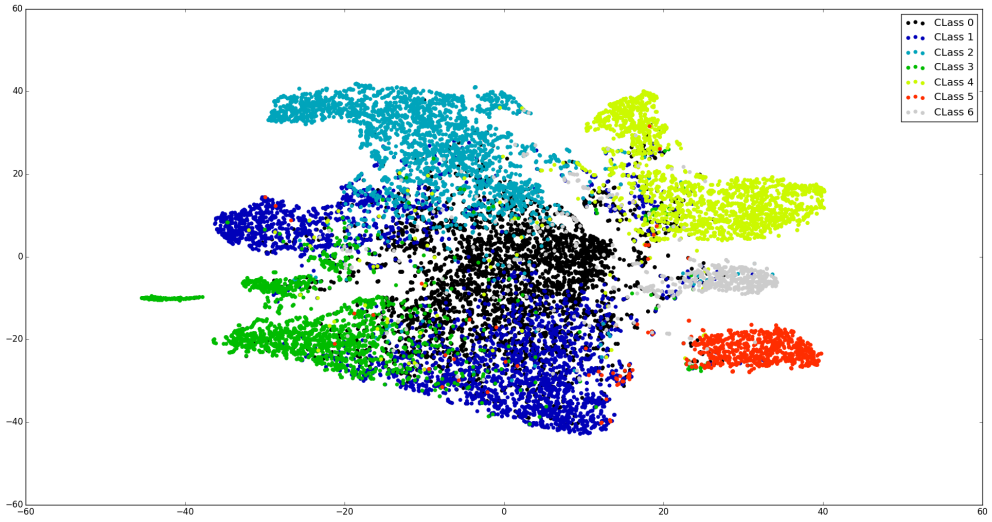



FIGURE 6.6: T-SNE embedding. (a) Embedding of MLC coral classes features with softmax loss after patch-wise training.

GoogleNet, 2. fully convolutional GoogleNet with Dense-CRF, 3. FCN-8s, and 4. FCN-8s with DCRF. Since the MLC dataset does not have any segmentation labels, we measured the performance of our system on the MLC dataset by evaluating its output at the test point locations. For the ADS dataset we created a expert labelled holdout test set of 10 images with segmentation labels. This let us report the segmentation performance using standard metrics of Intersection over Union(IoU) and mean pixel accuracy. For direct comparison, and to establish the benefit of localization during classification as hypothesized in subsection 6.6, we also report the classification accuracy over the test point locations in the ADS dataset as well.

## 6.8 Results

### 6.8.1 Patch recognition experiment

The overall accuracy(OA) of a an algorithm is defined as the sum of the number of correctly classified examples (the diagonal entries in the confusion matrix) divided by the total number of the tested examples. The extension of this evaluation criteria used specially for large scale object classification challenge [33, 34, 127] is the top-k accuracy, where k can be 5 or 10. We used top-2 accuracy since we have small class datasets.

TABLE 6.2: Summery of the experiments performed on ADS dataset.

| | VGG fine-tuned from ImageNet | GoogleNet fine-tuned from ImageNet | Training GoogleNet from Scratch |
|---|---|---|---|
| Iteration | 5500 | 20250 | 24500 ) |
| Training Error(start ⇒ end) | 2.4 ⇒ .25 | 4.6 ⇒ 1.3 | 3.7 ⇒ .27 |
| Test Error(start ⇒ end) | 2.8 ⇒ .35 | .55 ⇒ .68 | .72 ⇒ .6 |
| Top-1 (Top-2) accuracy(%) | **88.4 (97.1)** | 87.2 (95.5) | 84.1 (94.7) |

TABLE 6.3: Summery of the experiments conducted on MLC dataset.

| | VGG fine-tuned from ADS | GoogleNet fine-tuned from ImageNet | Training GoogleNet from Scratch |
|---|---|---|---|
| Iteration | 13250 | 21100 | 90000 |
| Training Error(start ⇒ end) | 2.04 ⇒ .47 | 3.27 ⇒ .5 | 4.3 ⇒ .69 |
| Test Error(start ⇒ end) | 1.96 ⇒ .45 | .79 ⇒ .46 | 1.3⇒.65 |
| Top-1 (Top-2) accuracy(%) | **85.2 (96.6)** | 83.5 (95.3) | 81.1 (94.2) |

***Classification results:*** Results of experiments on ADS is shown in Table 6.2 and the MLC in Table 6.3. On ADS, we achieve an overall accuracy of 84.1% top-1 and 94.7% top-2 score for the GoogleNet model trained from scratch and the training converges. While this accuracy and convergence rate are further improved (87.2% accuracy) if the weights are initialized from the Imagenet model. This shows the benefit of good weight initialization and also suggests that there may be many local minima in the weight space where the previous training got stuck. Hence for the VGG16 model we initialized the training from Imagenet weights from the beginning. The VGG16 model achieves an top-1 accuracy of 88.4% and top-2 accuracy of 97.1%. Both the networks achieve state-of-the-art classification result on the dataset.

Similarly for the MLC dataset the VGG16 fine-tuned from Imagenet weights achieves the best classification accuracy at 85.2% top-1 and 96.6% top-2. This is in comparison to the previous state-of-the art results reported by Beijbom *et al.* [142] on this dataset where they achieved an accuracy of 83.1%.

The detail results for MLC dataset are depicted using a normalized confusion matrix as shown in Figure 6.7. The classification works very well except for the

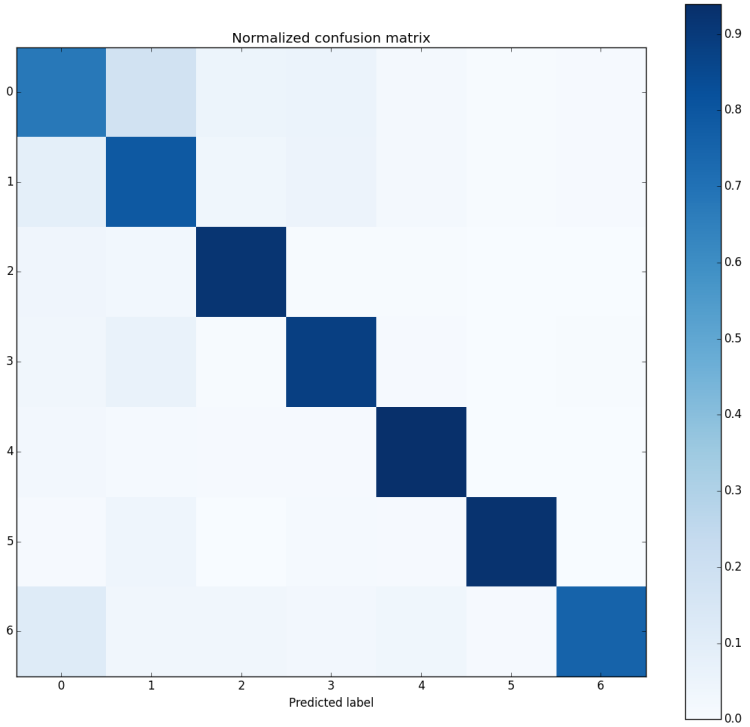



FIGURE 6.7: Confusion Matrix of the MLC dataset fine-tuned from each class number corresponds: 0-CCA, 1-Turf, 2-Sand, 3-Macro, 4-PORIT, 5-Pocill and 6-Monti.

two classes which described as ambiguously labelled in [142],i.e. the CCA and Turf classes. Since these two classes are the most frequent classes in the MLC dataset we believe that there might be a limit to the top-1 accuracy that can be achieved. This conclusion is further supported by the high top-2 accuracy obtained.

However to alleviate the limitations of patch based classification where defining the best patch size crop for each point is problematic, we decided to push the limits of the classification by localizing at the same time using the FCNN-DCRF framework. The results of which we report next.

### 6.8.2 Dense Classification results

We report classification accuracy on the test points for both MLC and ADS datasets. Furthermore, we also report mean pixel accuracy and Intersection over Union (IoU) results for the ADS dataset for the expert annotated segmentation test set. The classification accuracy is reported in Table 6.4. We compute the accuracy based on the point labels provided, i.e., we evaluate the output of our network on the true ground truth labels provided. It can be seen that classification accuracy

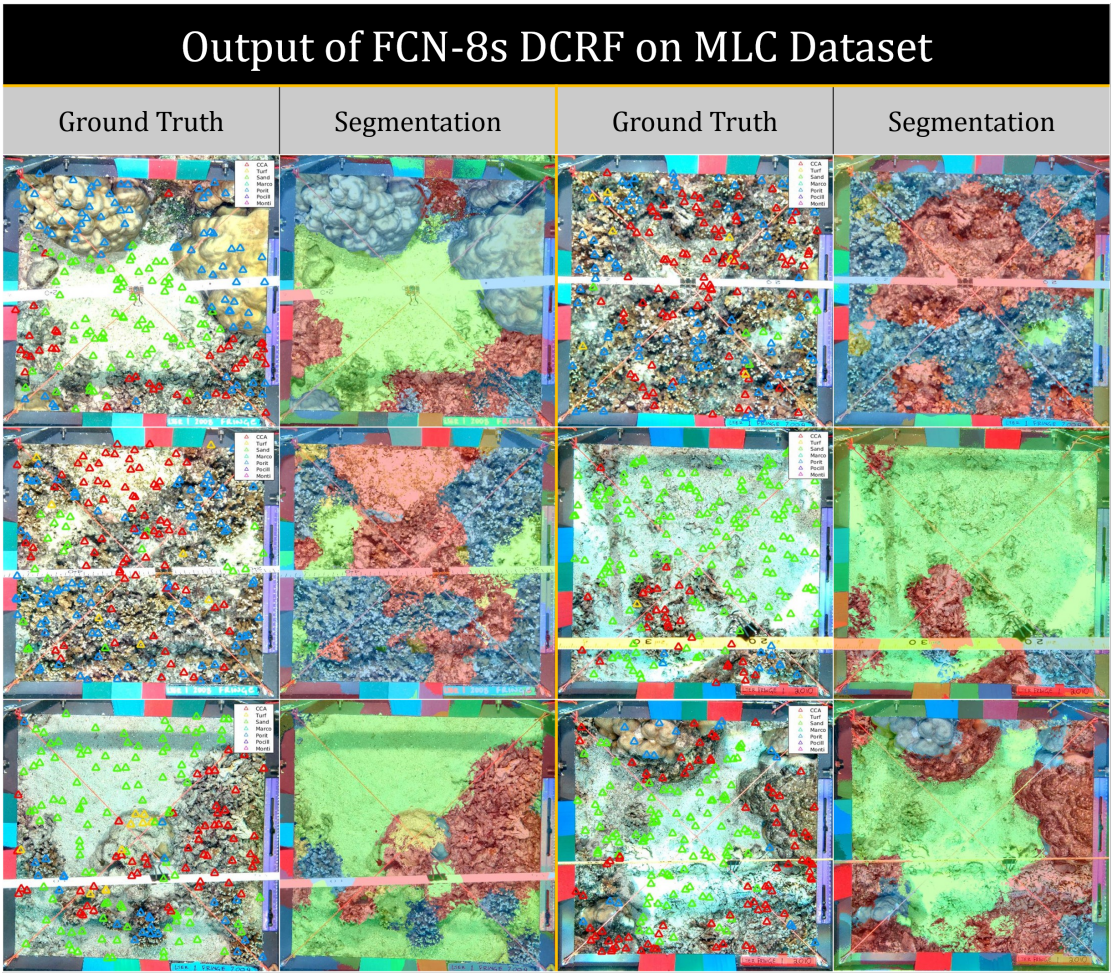



FIGURE 6.8: Illustrative output of our full system on the MLC dataset for dense classification trained with $\gamma = 5.5$

improves for every model. This validates our hypothesis that dense classification is better than patch extraction and classification. Furthermore, we observe that the DCRF does improve accuracy significantly in every case. For qualitative results we show the output of our system on the MLC dataset in Figure 6.8.

In Figure 6.9 we show the differentiation between a species of sponge and the Lophelia pertusa species of coral. In figure 6.10 we show the differentiation between live and dead Lophelia pertusa species of coral. It is noteworthy that these outputs are very significant in underwater mapping of flora and fauna.

### 6.8.3 Accuracy vs Speed

Our new integrated FCNN-DCRF surpassed human and other machine-learning performance benchmarks in speed and accuracy. Where expert annotation of

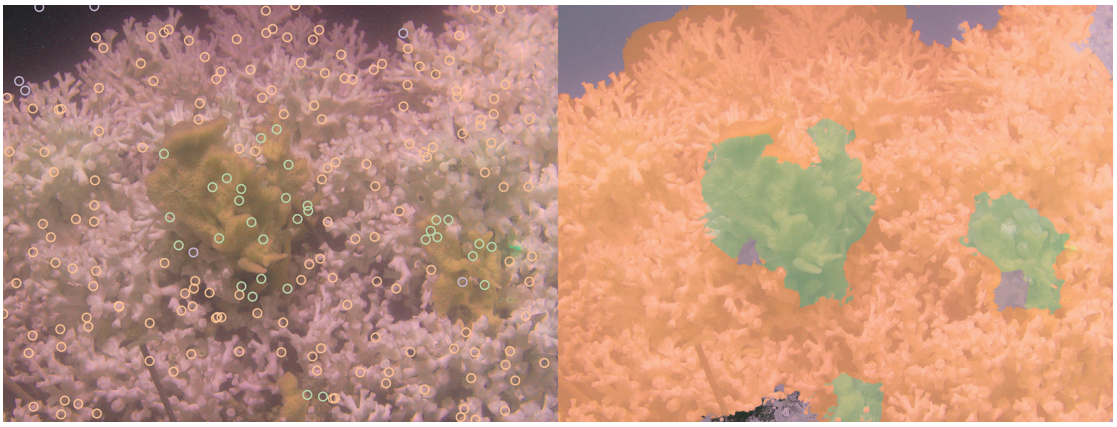


FIGURE 6.9: Segmentation showing live Lophelia pertusa coral and the sponge Mycale lingua.

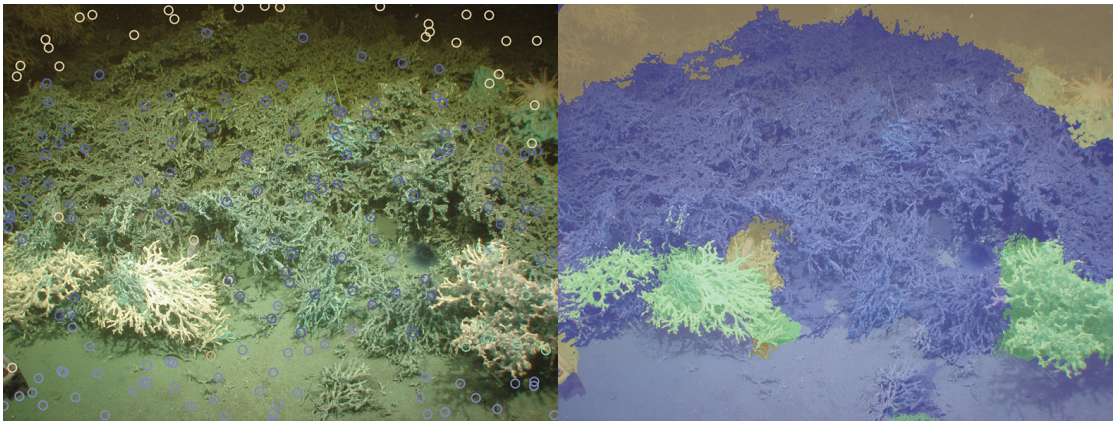


FIGURE 6.10: Segmentation of live versus dead Lophelia pertusa

TABLE 6.4: Classification accuracy after dense classification with $\gamma = 5.5$

| Model | Accuracy | |
|---|---|---|
| | MLC | ADS |
| Googlenet-Conv | 84.1 | 89.2 |
| Googlenet-Conv-DCRF | 85.6 | 91.2 |
| FCN-8s | 86.8 | 90.1 |
| FCN-8s-DCRF | **88.7** | **93.4** |

159 images took an average of 30 minutes per image, that's 286200 seconds for 159 images; the deep net takes less than 900 milliseconds for annotating each image. This represents a big jump in capabilities to do vision based ecological investigations that was not possible before.

### 6.8.4 Comparison to Expert Annotation

For completeness we gathered expert segmentation annotations on the ADS dataset. It must be noted that expert annotations are not 100% accurate due to the extreme tedious nature of the work. Especially in fine details, expert annotations are not very good. However for sake of comparison we compute intersection over union (IOU) metric between the automatic prediction and expert annotation. We achieve an mean IOU accuracy of 86.1% and pixel coverage accuracy of 91.5% . This shows that our method is very robust at dense prediction.

## 6.9 Conclusion

Advancement innovative platforms such as AUVs allows to collect large images and opens a door for machine learning researches in the area. The methods presented are based on latest developments in deep learning for object recognition application and are working for different applications. We showed how deep learning methods can be applied to small coral datasets by data augmentation using the off-the-shelf CNN architectures. The proposed method achieves overall classification accuracy of 88.7% on latest standard coral dataset, the MLC dataset. The methods is also tested on relatively smaller ADS coral dataset to achieve more than 93.4%.

Future work for the proposed method will cover preparing a standard coral dataset based on standards of deep learning datasets, making extensive augmentation using variety methods to improve over all result and developing unsupervised or semi-supervised deep learning methods also helps to use advantage available huge unlabelled data.

# Chapter 7

# Conclusion

*The purpose of this thesis was to develop a "A data driven machine learning approach can be adapted to solve the human head pose estimation problem. By using modern machine learning techniques we can solve the problem of detecting heads and estimating head pose in an unified way that spans multiple modalities (RGB and Depth) while being applicable to high resolution Human Machine Interface to low resolution surveillance domain simultaneously. The robust headpose estimation can then be used a signal for Social Signal Processing". To this end we show in Section 7.1 that we have met the objectives of the thesis. Furthermore, we showed wider applicability of our methods to the problem of deep underwater coral classification. In the following section we discuss future directions for research.*

## 7.1 Contributions

In this section we review our objectives from Chapter 1 and establish their link to the research presented. We show that the journey we embarked upon bore fruit in validating our hypothesis.

**Review current methods**

In Chapter 2 we reviewed all the relevant methods pertaining to head pose estimation. We also reviewed the methods pertaining to deep learning that we proposed to exploit for solving the problem. In Chapter 3 we took the state-of-the-art feature from literature (HOG) and proposed improvements (HDSC) that let us establish state-of-the-art performance. We compared the merits of the features quantitatively and showed that combining RGB and Depth information gave better overall results and the features were robust as well. Our proposal for the Zonal Kernel should be widely applicable to regression on closed manifolds.

**Bridge the gap in data requirements if any**

We identified that the headpose research literature lacks large scale quality data. Hence in Chapter 3 we introduced a high quality standard dataset that has the following properties.
(a) The data should be large scale for training deep neural networks.
(b) The data should have both RGB and Depth modalities.
(c) The data should span from very high to very low resolution.
(d) The data should capture annotations at a granularity of 1 degree or less to evaluate against regression approaches.
(e) Should cover all angles and not only frontal poses.

We successfully created such a dataset. Furthermore we characterized the error bounds of the headpose estimation approaches when eye tracking is not employed. That limit was found to be around 12.3°. That means our accuracies in many datasets are near the physical limit and in some cases surpasses that (although that does not contribute to the larger picture of social signal processing).

Finally we also annotated the large scale Hollywood head detection dataset with pose annotations. Thus we contributed significantly towards creating a unified

evaluation dataset that can be used for different modalities, detection, classification, regression.

### Exploit high-resolution to low-resolution imageries and multiple modalities

By creating our dataset that spans both HCI and surveillance domains and by mixing with various available datasets, we show in Chapter 4 that we can successfully learn from this data even in semi supervised settings by learning an underlying representation of human heads. We further show that we can learn a CNN from a RGB-D dataset and apply it successfully to surveillance domain problems.

### Method independent of explicit facial landmark detection

All the methods developed in Chapter 3 and 4 explicitly did not impose facial landmark tracking. However it should be noticed as shown in Chapter 4, certain CNN filters learned them implicitly anyway. However the CNN does not depend upon the presence of any such landmark. Hence, we eliminated the heuristic components of HCI head pose estimation algorithms.

### Do not require motion priors

As seen in prior work in low resolution surveillance data in Chapter 2, one way of achieving good accuracy in head pose estimation is to couple it to velocity estimate of the tracks. This is possible due to the fact that most people tend to look where they are going at any instant and that creates a bias in the dataset as seen in Figure 5.1. Hence that makes the information content of the pure headpose signal quite redundant. We explicitly avoided this while gathering the dataset by requiring subjects to move their head independent of body in any erratic fashion. Also all our methods developed in 4 do not even require any temporal smoothing. Hence the information content of the headpose signal is not attenuated at all. This can be seen from the performance when applied to Social signal processing.

### Create unified end-to-end detection and headpose estimation framework

In Chapter 4.3 we created a novel architecture and loss function that solves the problem of detection and pose estimation in an end-to-end fashion. The intuition behind any end-to-end solution is that the joint optimisation of the objectives promote symbioses among the tasks. However one must be careful as shown in Chapter 2 because the tasks might be orthogonal. However with careful consideration we developed an architecture that proved to be state of the art.

**Evaluate against public datasets**

We evaluated all our approaches on public datasets against competing methods from literature. In Chapter 3 we evaluated against the baseline methods and showed improved performance. In Chapter 4 we reimplemented most methods and tested on multiple datasets like the Oxford, Caviar, Multi-PIE, Biwi Kinect and our own data. We also compared our DFCNN detection and pose estimation against the best detector and pose estimator on the Hollywood heads dataset. On all datasets, apart from detection on Hollywood heads we reported the state-of-the-art approach. We performed slightly worse in detection on the Hollywood head dataset, but we believe that that dataset strongly favours the competing Contextual CNN method as it takes into account spatial relationships among heads. Movie scenes have structures that are learn-able that way. However in an unstructured scene the Oxford dataset, our detector significantly outperforms other methods. Hence we conclude that our methods are very robust and validated against other methods on available datasets.

**Applicable to social signal processing**

Finally, in Chapter 5 we showed that by having a strong head pose signal and novel Attention and Interaction Metrics, we significantly outperform other methods like Here's Looking at you Kid on social signal datasets. We also showed that our headpose estimation was successful in predicting pedestrian behaviours, thereby improving performance of a Kalman filter tracker with intentional priors.

**Find general applicability beyond the Headpose problem**

We defined and used the ALICE loss for the end-to-end head detection and head-pose classification. However, with very little modification it was shown to be applicable to sparse point based annotation problems. We established that by

applying it to solve the challenging problem of automatic segmentation of deep sea corals in Chapter 6.

Although there are many ways to take the work forward, we conclude that we have successfully completed the exploration we set out for in the beginning. However we discuss some ideas of how the research enables exploration of new ideas briefly in the next section.

To achieve these objectives, we employed a data driven machine learning approach as can be seen from Chapter 3 and 4. To this end, having reached the objectives by employing the methods hypothesised in the thesis statement as stated in 1, we can conclude the thesis to be true and valid.

**Find general applicability beyond the Headpose problem** We defined and used the ALICE loss for the end-to-end head detection and headpose classification. However, with very little modification it was shown to be applicable to sparse point based annotation problems. We showed this by

## 7.2 Future Work

In this section we discuss potential future directions of this work.

**Higher level Social Signal Processing**
Although we have explored some aspects of social signal processing, we believe a lot of more research and standardised datasets are required in the field. We need to understand group behaviour from headpose data. Similarly, it will be interesting to apply headpose to separate communicative gestures from general actions. That will then lead to fine grained gesture classifications into various gesture phases like pre-stoke and stroke as hypothesised in [6]. Once we have large scale standard datasets for human gestures, headpose and other low level signals like limb movement and gesture mimicry can be used for better understanding the behaviour of humans or groups of humans.

**End-to-end detection and pose estimation of other objects**
The DFCNN framework can be used for other tasks like bodypose estimation. The

ALICE loss lends itself to learn from weak labels. Hence body parsing part segmentation can be achieved from body joint datasets like FLIC and Leeds Sports. Other applications might include combining the output with recurrent neural networks for action/gesture recognition and distinction via head pose. It can be applied for pose classification problems for other objects like cars for predicting traffic from instantaneous snapshots.

**Better framework and understanding of early feature fusion of different modalities in Deep neural network**

So far there does not seem to be a theory of multi modal convolutional neural networks that fuse features early for joint optimisation. It has been speculated that one source of gradient is not meaningful for different modalities. It has also been seen from our work in Chapter 4 that this recipe does not yield good results. However theoretical investigation is required to understand this phenomenon better so that multiple modalities can be jointly optimised.

## 7.3 Final Remarks

In this thesis we set out to solve the problem of human headpose because it is one of the most important underlying signals for Social Signal Processing. We successfully identified the problems with current approaches and we proposed new methods and solved the problem in a unified manner for both detection and pose estimation in multi-modal RGB-D data. We showed the robustness of the developed algorithms in both HCI and surveillance domains. Beyond that we showed that the algorithms are generally applicable by applying them to Deep sea coral segmentation. We achieved the state of the art accuracy in all domains for headpose estimation, and showed that by having an informative signal, it is applicable to higher lever social signal processing that is very useful for human behaviour understanding. The main message of this thesis is that a data driven machine learning based strong and robust solution to an underlying signal processing subproblem (head pose) can lead to advanced applications that will one day help computers understand subtleties of human behaviour.

# Bibliography


[1] G. Fanelli, J. Gall, and L. Van Gool. Real time head pose estimation with random regression forests. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 617–624, June 2011. doi: 10.1109/CVPR.2011.5995458.

[2] Ralph Gross, Iain Matthews, Jeffrey Cohn, Takeo Kanade, and Simon Baker. Multi-pie. *Image and Vision Computing*, 28(5):807–813, 2010.

[3] Tuan-Hung Vu, Anton Osokin, and Ivan Laptev. Context-aware cnns for person head detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2893–2901, 2015.

[4] Ben Benfold and Ian Reid. Colour invariant head pose classification in low resolution video. In *Proceeding of the British Machine Vision Conference*, 2008.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[6] Adam Kendon. *Gesture: Visible action as utterance*. Cambridge University Press, 2004.

[7] Stephen RH Langton, Helen Honeyman, and Emma Tessler. The influence of head contour and nose angle on the perception of eye-gaze direction. *Perception & psychophysics*, 66(5):752–771, 2004.

[8] Rolf H Baxter, Michael JV Leach, Sankha S Mukherjee, and Neil M Robertson. An adaptive motion model for person tracking with instantaneous head-pose features. *Signal Processing Letters, IEEE*, 22(5):578–582, 2015.

[9] Neil Robertson and Ian Reid. Estimating gaze direction from low-resolution faces in video. In *Computer Vision–ECCV 2006*, pages 402–415. Springer, 2006.

[10] Ben Benfold and Ian Reid. Colour invariant head pose classification in low resolution video. In *BMVC*, pages 1–10, 2008.

[11] Cheng Chen and Jean-Marc Odobez. We are not contortionists: coupled adaptive learning for head and body orientation estimation in surveillance video. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1544–1551. IEEE, 2012.

[12] Teera Siriteerakul, Yoichi Sato, and Veera Boonjing. Estimating change in head pose from low resolution video using lbp-based tracking. In *Intelligent Signal Processing and Communications Systems (ISPACS), 2011 International Symposium on*, pages 1–6. IEEE, 2011.

[13] Gabriele Fanelli, Juergen Gall, and Luc Van Gool. Real time head pose estimation with random regression forests. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 617–624. IEEE, 2011.

[14] Kenneth Alberto Funes Mora and Jean-Marc Odobez. Gaze estimation from multimodal kinect data. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 25–30. IEEE, 2012.

[15] Pashalis Padeleris, Xenophon Zabulis, Antonis Argyros, et al. Head pose estimation on depth data based on particle swarm optimization. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 42–49. IEEE, 2012.

[16] Sankha S Mukherjee and Neil M Robertson. Deep head pose: Gaze-direction estimation in multimodal video. *IEEE Transactions on Multimedia*, 17(11): 2094–2107, 2015.

[17] Sankha S. Mukherjee, Rolf H. Baxter, and Neil M. Robertson. Watch where you're going! - pedestrian tracking via head pose. In *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pages 573–579, 2016. ISBN 978-989-758-175-5. doi: 10.5220/0005786905730579.

[18] Sankha S Mukherjee, Rolf H Baxter, and Neil M Robertson. Instantaneous real-time head pose at a distance. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 3471–3475. IEEE, 2015.

[19] Henry Lea-Ann, Sankha S Mukherjee, Neil M Robertson, Laurence De Clippele, and J. Murray Roberts. Deep corals, deep learning: Moving the deep net towards real-time image annotation. In *6th International Symposium on Deep-Sea Corals*. Harvard, 2016.

[20] Teera Siriteerakul, Daisuke Sugimura, and Yoichi Sato. Head pose classification from low resolution images using pairwise non-local intensity and color differences. In *Image and Video Technology (PSIVT), 2010 Fourth Pacific-Rim Symposium on*, pages 362–369. IEEE, 2010.

[21] Nicolas Gourier, Jérôme Maisonnasse, Daniela Hall, and James L Crowley. Head pose estimation on low resolution images. In *Multimodal Technologies for Perception of Humans*, pages 270–280. Springer, 2007.

[22] Rainer Stiefelhagen. Estimating head pose with neural networks-results on the pointing04 icpr workshop evaluation data. In *Pointing 04 ICPR Workshop of the Int. Conf. on Pattern Recognition*, 2004.

[23] Vineeth Nallure Balasubramanian, Jieping Ye, and Sethuraman Panchanathan. Biased manifold embedding: A framework for person-independent head pose estimation. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–7. IEEE, 2007.

[24] Chiraz BenAbdelkader. Robust head pose estimation using supervised manifold learning. In *Computer Vision–ECCV 2010*, pages 518–531. Springer, 2010.

[25] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[26] Dario Cazzato, Marco Leo, and Cosimo Distante. An investigation on the feasibility of uncalibrated and unconstrained gaze tracking for human assistive applications by using head pose estimation. *Sensors*, 14(5):8363–8379, 2014.

[27] P. Padeleris, X. Zabulis, and A.A. Argyros. Head pose estimation on depth data based on particle swarm optimization. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 42–49, 2012.

[28] Manuel J Marín-Jiménez, Andrew Zisserman, and Vittorio Ferrari. Heres looking at you, kid. detecting people looking at each other in videos. In *British Machine Vision Conference*, 2011.

[29] Qicong Wang, Yuxiang Wu, Yehu Shen, Yong Liu, and Yunqi Lei. Supervised sparse manifold regression for head pose estimation in 3d space. *Signal Processing*, 112:34–42, 2015.

[30] Xi Peng, Junzhou Huang, Qiong Hu, Shaoting Zhang, Ahmed Elgammal, and Dimitris Metaxas. From circle to 3-sphere: Head pose estimation by instance parameterization. *Computer Vision and Image Understanding*, 136: 92–102, 2015.

[31] Bingpeng Ma, Annan Li, Xiujuan Chai, and Shiguang Shan. Covga: A novel descriptor based on symmetry of regions for head pose estimation. *Neurocomputing*, 143:97–108, 2014.

[32] ByungOk Han, Suwon Lee, and Hyun S Yang. Head pose estimation using image abstraction and local directional quaternary patterns for multiclass classification. *Pattern Recognition Letters*, 45:145–153, 2014.

[33] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.

[34] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[35] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 29(6):82–97, 2012.

[36] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Comput.*, 18(7):1527–1554, July 2006. ISSN 0899-7667. doi: 10.1162/neco.2006.18.7.1527. URL http://dx.doi.org/10.1162/neco.2006.18.7.1527.

[37] Yichuan Tang, Ruslan Salakhutdinov, and Geoffrey Hinton. Robust boltzmann machines for recognition and denoising. In *IEEE Conference on Computer Vision and Pattern Recognition, 2012, Providence, Rhode Island, USA*, 2012.

[38] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going Deeper with Convolutions. *ArXiv e-prints*, September 2014.

[39] Jia Deng, Kai Li, Minh Do, Hao Su, and Li Fei-Fei. Construction and analysis of a large scale image ontology. *Vision Sciences Society*, 186, 2009.

[40] Luís A Alexandre. 3d object recognition using convolutional neural networks with transfer learning between input channels. In *Proc. the 13th International Conference on Intelligent Autonomous Systems*, 2014.

[41] Saurabh Gupta, Ross Girshick, Pablo Arbeláez, and Jitendra Malik. Learning rich features from rgb-d images for object detection and segmentation. In *Computer Vision–ECCV 2014*, pages 345–360. Springer, 2014.

[42] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *arXiv preprint arXiv:1411.4038*, 2014.

[43] Steve Lawrence, C Lee Giles, Ah Chung Tsoi, and Andrew D Back. Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, 8(1):98–113, 1997.

[44] Patrice Y Simard, David Steinkraus, and John C Platt. Best practices for convolutional neural networks applied to visual document analysis. In *ICDAR*, volume 3, pages 958–962, 2003.

[45] Fu Jie Huang and Yann LeCun. Large-scale learning with svm and convolutional for generic object categorization. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 284–291. IEEE, 2006.

[46] Dan C Ciresan, Ueli Meier, Jonathan Masci, Luca Maria Gambardella, and Jürgen Schmidhuber. Flexible, high performance convolutional neural networks for image classification. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 1237, 2011.

[47] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[48] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[49] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1): 98–136, 2015.

[50] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014*, pages 818–833. Springer, 2014.

[51] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[52] Anne-Marie Tousch, Stéphane Herbin, and Jean-Yves Audibert. Semantic hierarchies for image annotation: A survey. *Pattern Recognition*, 45(1):333–345, 2012.

[53] Nitish Srivastava and Ruslan R Salakhutdinov. Discriminative transfer learning with tree-based priors. In *Advances in Neural Information Processing Systems*, pages 2094–2102, 2013.

[54] Zhenhua Wang, Xingxing Wang, and Gang Wang. Learning fine-grained features via a cnn tree for large-scale classification. *arXiv preprint arXiv:1511.04534*, 2015.

[55] Tianjun Xiao, Jiaxing Zhang, Kuiyuan Yang, Yuxin Peng, and Zheng Zhang. Error-driven incremental learning in deep convolutional neural network for large-scale image classification. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 177–186. ACM, 2014.

[56] Zhicheng Yan, Vignesh Jagadeesh, Dennis Decoste, Wei Di, and Robinson Piramuthu. Hd-cnn: hierarchical deep convolutional neural network for image classification. In *International Conference on Computer Vision (ICCV)*, volume 2, 2015.

[57] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *Proc. CVPR Workshop on Fine-Grained Visual Categorization (FGVC)*, volume 2, 2011.

[58] Linjie Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. A large-scale car dataset for fine-grained categorization and verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3973–3981, 2015.

[59] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Computer Vision, Graphics & Image Processing, 2008. ICVGIP'08. Sixth Indian Conference on*, pages 722–729. IEEE, 2008.

[60] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[61] Thomas Berg, Jiongxin Liu, Seung Woo Lee, Michelle L Alexander, David W Jacobs, and Peter N Belhumeur. Birdsnap: Large-scale fine-grained visual categorization of birds. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2019–2026. IEEE, 2014.

[62] Steve Branson, Grant Van Horn, Serge Belongie, and Pietro Perona. Improved bird species categorization using pose normalized deep convolutional nets. *arXiv preprint arXiv:1406.2952*, 2014.

[63] Ning Zhang, Jeff Donahue, Ross Girshick, and Trevor Darrell. Part-based r-cnns for fine-grained category detection. In *European Conference on Computer Vision*, pages 834–849. Springer, 2014.

[64] Jasper RR Uijlings, Koen EA van de Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[65] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1653–1660, 2014.

[66] Ben Sapp and Ben Taskar. Modec: Multimodal decomposable models for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3674–3681, 2013.

[67] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *BMVC*, volume 2, page 5, 2010.

[68] Yi Yang and Deva Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1385–1392. IEEE, 2011.

[69] Fang Wang and Yi Li. Beyond physical connections: Tree models in human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 596–603, 2013.

[70] Leonid Pishchulin, Mykhaylo Andriluka, Peter Gehler, and Bernt Schiele. Poselet conditioned pictorial structures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 588–595, 2013.

[71] Jonathan J Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *Advances in neural information processing systems*, pages 1799–1807, 2014.

[72] Xianjie Chen and Alan L Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *Advances in Neural Information Processing Systems*, pages 1736–1744, 2014.

[73] Xianjie Chen and Alan L Yuille. Parsing occluded people by flexible compositions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3945–3954, 2015.

[74] Tomas Pfister, James Charles, and Andrew Zisserman. Flowing convnets for human pose estimation in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1913–1921, 2015.

[75] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[76] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European Conference on Computer Vision*, pages 346–361. Springer, 2014.

[77] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[78] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[79] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *arXiv preprint arXiv:1512.00567*, 2015.

[80] Russell Stewart and Mykhaylo Andriluka. End-to-end people detection in crowded scenes. *arXiv preprint arXiv:1506.04878*, 2015.

[81] Lijun Wang, Wanli Ouyang, Xiaogang Wang, and Huchuan Lu. Visual tracking with fully convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3119–3127, 2015.

[82] Yan Chen, Xiangnan Yang, Bineng Zhong, Shengnan Pan, Duansheng Chen, and Huizhen Zhang. Cnntracker: Online discriminative object tracking via deep convolutional neural network. *Applied Soft Computing*, 38:1088–1098, 2016.

[83] Hanxi Li, Yi Li, and Fatih Porikli. Deeptrack: Learning discriminative feature representations online for robust visual tracking. *IEEE Transactions on Image Processing*, 25(4):1834–1848, 2016.

[84] N Johnson and Dc Hogg. Learning the Distribution of Object Trajectories for Event Recognition. *Image and Vision Computing*, 14:583–592, 1996. doi: 10.5244/C.9.58. URL http://www.bmva.org/bmvc/1995/bmvc-95-057.html.

[85] C. Stauffer and W.E.L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, 2000. ISSN 01628828. doi: 10.1109/34.868677. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=868677.

[86] Weiming Hu, Xuejuan Xiao, Zhouyu Fu, Dan Xie, Tieniu Tan, and Steve Maybank. A system for learning statistical motion patterns. *IEEE transactions on pattern analysis and machine intelligence*, 28(9):1450–64, September 2006. ISSN 0162-8828. doi: 10.1109/TPAMI.2006.176. URL http://www.ncbi.nlm.nih.gov/pubmed/16929731.

[87] B.T. Morris and M.M. Trivedi. A Survey of Vision-Based Trajectory Learning and Analysis for Surveillance. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(8):1114–1127, August 2008. ISSN 1051-8215. doi: 10.1109/TCSVT.2008.927109. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4543858.

[88] Dimitios Makris and Tim Ellis. Learning semantic scene models from observing activity in visual surveillance. *Systems, Man, and Cybernetics, Part B*, 35(3):397–408, 2005.

[89] C. Piciarelli and G.L. Foresti. On-line trajectory clustering for anomalous events detection. *Pattern Recognition Letters*, 27(15):1835–1842, November 2006. ISSN 01678655. doi: 10.1016/j.patrec.2006.02.004. URL http://linkinghub.elsevier.com/retrieve/pii/S0167865506000432.

[90] Zia Khan, Tucker Balch, and Frank Dellaert. MCMC-based particle filtering for tracking a variable number of interacting targets. *IEEE transactions on pattern analysis and machine intelligence*, 27(11):1805–19, November 2005. ISSN 0162-8828. doi: 10.1109/TPAMI.2005.223. URL http://www.ncbi.nlm.nih.gov/pubmed/16285378.

[91] S Pellegrini, A Ess, K Schindler, and L van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th International Conference on Computer Vision*, pages 261–268. Ieee, September 2009. ISBN 978-1-4244-4420-5. doi: 10.1109/ICCV.2009.5459260. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5459260.

[92] Julio Cezar Silveira Jacques, Adriana Braun, John Soldera, Soraia Raupp Musse, and Cláudio Rosito Jung. Understanding people motion in video sequences using Voronoi diagrams. *Pattern Analysis and Applications*, 10 (4):321–332, April 2007. ISSN 1433-7541.

[93] Stefano Pellegrini and Luc Van Gool. Tracking with a mixed continuous-discrete Conditional Random Field. *Computer Vision and Image Understanding*, 117(10):1215–1228, October 2013. ISSN 10773142.

[94] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning hierarchical features for scene labeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1915–1929, 2013.

[95] Camille Couprie, Clément Farabet, Laurent Najman, and Yann LeCun. Indoor semantic segmentation using depth information. *arXiv preprint arXiv:1301.3572*, 2013.

[96] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv:1606.00915*, 2016.

[97] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip Torr. Conditional random fields as recurrent neural networks. In *International Conference on Computer Vision (ICCV)*, 2015.

[98] George Papandreou, Liang-Chieh Chen, Kevin Murphy, and Alan L Yuille. Weakly- and semi-supervised learning of a dcnn for semantic image segmentation. *arxiv:1502.02734*, 2015.

[99] N.M. Robertson and I.D. Reid. Estimating gaze direction from low-resolution faces in video. In *Proceeding of the 9th European Conference on Computer Vision, 2006*, volume 3952/2006, pages 402–415, 2006.

[100] Chen Cheng and J. Odobez. We are not contortionists: Coupled adaptive learning for head and body orientation estimation in surveillance video. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1554–1551, 2012.

[101] T. Siriteerakul, Y. Sato, and V. Boonjing. Estimating change in head pose from low resolution video using lbp-based tracking. In *Proceeding of the Intelligent Signal Processing and Communication System*, 2011.

[102] K.A. Funes Mora and J. Odobez. Gaze estimation from multimodal kinect data. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 25–30, 2012.

[103] N. Gourier, J. Maisonnasse, D. Hall, and J.L. Crowley. Head pose estimation on low resolution images. In *Proceeding of the 1st International Evaluation Conference on Classification of Events, Activities and Relationships*, pages 270–280, 2006.

[104] R. Stiefelhagen. Estimating head pose with neural network-results on the pointing04 icpr workshop evaluation data. In *Proceedings of the ICPR Workshop on Visual Observation of Deictic Gestures*, 2004.

[105] V. Balasubramanian, J. Ye, and S. Panchanathan. Biased manifold embedding: a framework for person-independent head pose estimation. In *Proceeding of the IEEE Conference on Computer Vision and PatternRecognition*, pages 1–7, 2007.

[106] C. BenAbdelkader. Robust head pose estimation using supervised manifold learning. In *Proceeding of the 11th European Conference on Computer Vision*, pages 518–531, 2010.

[107] M. J. Marin-Jimenez, A. Zisserman, and V. Ferrari. Here's looking at you kid. detecting people looking at each other in videos. In *British Machine Vision Conference (BMVC)*, 2011.

[108] Alexander Belyaev. Implicit image differentiation and filtering with applications to image sharpening. *SIAM Journal on Imaging Sciences*, 6(1): 660–679, 2013.

[109] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005. ISBN 026218253X.

[110] Diego Tosato, Mauro Spera, Matteo Cristani, and Vittorio Murino. Characterizing humans on riemannian manifolds. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):1972–1984, 2013.

[111] Gabriele Fanelli, Matthias Dantone, Juergen Gall, Andrea Fossati, and Luc Van Gool. Random forests for real time 3d face analysis. *International Journal of Computer Vision*, 101(3):437–458, 2013.

[112] C. Vondrick, A. Khosla, T. Malisiewicz, and A. Torralba. HOGgles: Visualizing Object Detection Features. *ICCV*, 2013.

[113] Shuai Tang, Xiaoyu Wang, Xutao Lv, TonyX Han, James Keller, Zhihai He, and Marjorie Skubic. Histogram of oriented normal vectors for object recognition with a depth sensor. In *Lecture Notes in Computer Science*, volume 7725, pages 525–538. Springer Berlin Heidelberg, 2013.

[114] G. E. Fasshauer. Positive definite kernels: past, present and future. *Dolomites Research Notes on Approximation*, 4(Special Issue on Kernel Functions and Meshless Methods):21–63, 2011. URL http://drna.di.univr.it/volume04.html.

[115] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194. ACM Press/Addison-Wesley Publishing Co., 1999.

[116] Ben Benfold and Ian Reid. Unsupervised learning of a scene-specific coarse gaze estimator. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2344–2351, 2011.

[117] Zhanpeng Zhang, Ping Luo, ChenChange Loy, and Xiaoou Tang. Facial landmark detection by deep multi-task learning. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision - ECCV 2014*, volume 8694 of *Lecture Notes in Computer Science*, pages 94–108. Springer International Publishing, 2014. ISBN 978-3-319-10598-7.

[118] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[119] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[120] Alex Krizhevsky. One weird trick for parallelizing convolutional neural networks. *arXiv preprint arXiv:1404.5997*, 2014.

[121] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *arXiv preprint arXiv:1502.01852*, 2015.

[122] Ben Benfold and Ian Reid. Unsupervised learning of a scene-specific coarse gaze estimator. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2344–2351. IEEE, 2011.

[123] Sanjiva K Lele. Compact finite difference schemes with spectral-like resolution. *Journal of computational physics*, 103(1):16–42, 1992.

[124] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678. ACM, 2014.

[125] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010.

[126] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[127] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*, 2014.

[128] Gunilla Borgefors. Distance transformations in digital images. *Computer vision, graphics, and image processing*, 34(3):344–371, 1986.

[129] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[130] R E Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[131] Greg Welch and Gary Bishop. An Introduction to the Kalman Filter. Technical report, Department of Computer Science, University of North Carolina, 2006.

[132] Rolf Hugh Baxter, Michael Leach, and Neil M. Robertson. Tracking with Intent. In *Sensor Signal Prcoessing for Defence*, 2014.

[133] Edinburgh University Informatics Department. CAVIAR: Context Aware Vision using Image-based Active Recognition.

[134] Kanti V Mardia and Peter E Jupp. *Directional statistics*, volume 494. John Wiley & Sons, 2009.

[135] Steven M Boker, Jennifer L Rotondo, Minquan Xu, and Kadijah King. Windowed cross-correlation and peak picking for the analysis of variability in the association between behavioral time series. *Psychological Methods*, 7(3):338, 2002.

[136] Katie Reytar Lauretta Burke. Reefs at risk revisited in the coral triangle @ONLINE. http://www.wri.org/publication/reefs-risk-revisited-coral-triangle, July 2012. [Online; accessed 19-May-2015].

[137] Steve Branson, Grant Van Horn, Serge Belongie, and Pietro Perona. Bird species categorization using pose normalized deep convolutional nets. *arXiv preprint arXiv:1406.2952*, 2014.

[138] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pages 512–519. IEEE, 2014.

[139] Ma Shiela Angeli Marcos, Maricor Soriano, and Caesar Saloma. Classification of coral reef images from underwater video using neural networks. *Optics express*, 13(22):8766–8771, 2005.

[140] Anand Mehta, Eraldo Ribeiro, Jessica Gilner, and Robert van Woesik. Coral reef texture classification using support vector machines. In *VISAPP (2)*, pages 302–310, 2007.

[141] M Dale Stokes and Grant B Deane. Automated processing of coral reef benthic images. *Limnology and Oceanography: Methods*, 7(2):157–168, 2009.

[142] Oscar Beijbom, Peter J Edmunds, David I Kline, B Greg Mitchell, and David Kriegman. Automated annotation of coral reef survey images. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1170–1177. IEEE, 2012.

[143] Eduardo Tusa, Alan Reynolds, David M Lane, Neil M Robertson, Hyxia Villegas, and Antonio Bosnjak. Implementation of a fast coral detector using a supervised machine learning and gabor wavelet feature descriptors. In *Sensor Systems for a Changing Ocean (SSCO), 2014 IEEE*, pages 1–6. IEEE, 2014.

[144] ASM Shihavuddin, Nuno Gracias, Rafael Garcia, Arthur CR Gleason, and Brooke Gintert. Image-based coral reef classification and thematic mapping. *Remote Sensing*, 5(4):1809–1841, 2013.

[145] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[146] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[147] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International conference on artificial intelligence and statistics*, pages 249–256, 2010.

[148] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*, 2011.

[149] Andrew Adams, Jongmin Baek, and Myers Abraham Davis. Fast high-dimensional filtering using the permutohedral lattice. In *Computer Graphics Forum*, volume 29, pages 753–762. Wiley Online Library, 2010.

[150] Mattis Paulin, Jérôme Revaud, Zaid Harchaoui, Florent Perronnin, and Cordelia Schmid. Transformation pursuit for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3646–3653. IEEE, 2014.

[151] Keven Kedao Wang. Image classification with pyramid representation and rotated data augmentation on torch 7.

[152] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015. URL http://arxiv.org/abs/1412.7062.